

THÈSE

Soutenue le 23 juin 1980 pour obtenir le

**DIPLÔME DE DOCTEUR EN MÉTHODES D'APPROXIMATION
ET ALGORITHMES EN ANALYSE ET THÉORIE DES NOMBRES**

par

Sadeq EL-HAMOUZ

★
1980

**ORDONNANCEMENT DE PRODUCTION SUR DEUX
TRAITEMENTS SUCCESSIFS AVEC TEMPS
DE TRANSITION**

Devant la Commission d'examen :

M. MIELLOU J.C.

Président

MM. GOIFFON
LEMAIRE
ROUBELLAT
TREHEL

}
Examineurs

Digitized by the Internet Archive
in 2026

Laboratoire d'Analyse
Numérique et Informatique

Université de Franche-Comté
Faculté des sciences et des techniques

THÈSE

Soutenue le 23 juin 1980 pour obtenir le

**DIPLOME DE DOCTEUR EN METHODES D'APPROXIMATION
ET ALGORITHMES EN ANALYSE ET THEORIE DES NOMBRES**

par

Sadeq EL-HAMOUZ

★

**ORDONNANCEMENT DE PRODUCTION SUR DEUX
TRAITEMENTS SUCCESSIFS AVEC TEMPS
DE TRANSITION**

Devant la Commission d'examen :

M. MIELLOU J.C.

Président

MM. GOIFFON
LEMAIRE
ROUBELLAT
TREHEL

}

Examineurs

THÈSE

Présentée le 25 juin 1980 pour obtenir le

DIPLOME DE DOCTEUR EN MÉTHODES D'APPROXIMATION
ET ALGORITHMES EN ANALYSE ET THÉORIE DES NOMBRES

Par M. J. J. VAN DER WOUDE



ORDONNANCEMENT DE PRODUCTION SUR DEUX
TRAITEMENTS SUCCESSIFS AVEC TEMPS
DE TRANSITION

1980

Document de l'Université de Leiden

Présenté

par M. J. J. VAN DER WOUDE

Présenté

M. J. J. VAN DER WOUDE
Lecteur
Bibliothèque
Université de Leiden

UNIVERSITÉ DE FRANCHE-COMTÉ BESANÇON

Président : Monsieur le Professeur Pierre LÉVÊQUE

Président honoraire : Monsieur le Professeur Jean THIÉBAUT

FACULTÉ DES SCIENCES ET DES TECHNIQUES

DOYEN

M. le Professeur Jean BULABOIS
Physique Générale-optique

ASSESSEURS

M. le Professeur Bernard MILLET
Botanique

M. Robert GUYETANT, Maître-Assistant
Biol. animale Ecologie

DOYENS HONORAIRES

MM. GLANGEAUD, JACQUEMAIN, CHATELET,
THIEBAUT, BERNARD

PROFESSEURS HONORAIRES

MM. BENNETON, CHATELET, MESNAGE, QUANTIN
THEOBALD N., THIRY, TRILLAT, TRONCHET,
UEBERSFELD

MM. BANTEGNIE Robert	: Mathématiques
BENILAN Philippe	: Mathématiques
BERNARD Jean	: Chimie Physique
BIDAULT Michel	: Taxonomie expérimentale
BROQUET Paul	: Géologie
BRUCKERT Sylvain	: Ecologie végétale
BULABOIS Jean	: Physique Gle optique
CAPODANNO Pierre	: Mécanique théorique
CERUTTI Ernest	: Chimie appliquée
CHALEAT Raymond	: Mécanique appliquée
CHARDON J. Claude	: Spectroscopie
CHAUVE Pierre	: Géologie minéralogie
COUGNARD Jean	: Mathématiques
DEVIN Claude	: Chimie 1er cycle
DUBOUCHET Jacques	: Physiologie végétale
GALATRY Louis	: Physique moléculaire
GAUDEMER Yves	: Biochimie
GOMOT Lucien	: Zoologie-Embryologie
GOUARNE René	: Mathématiques
GRAS Georges	: Mathématiques
GREMILLARD Jean	: Mathématiques
KARCHE J. Paul	: Pétrographie Minéralogie
LALLEMENT Gérard	: Mécanique appliquée
LAUDE Bernard	: Chimie organique
LESAINTE Pierre	: Mathématiques I
LHOTE François	: Automatique
MERIGOUX Henri	: Cristallographie et Synthèses minérales

MIELLOU J. Claude	: Analyse numérique
MILLET Bernard	: Botanique
MONTAGNER Hubert	: Psychophysiologie
MOREELS Guy	: Astronomie
OLIVIER Marcel	: Electroniq. quantique
PARIZET Jean	: Mathématiques
PERROT Roger	: Chimie générale
PLUVINAGE Philippe	: Physique théorique
POTIER Robert	: Physique des sols
RANGHEARD Yves	: Géologie
REAL Pierre	: Ecologie animale
RIPPLINGER Jean	: Physiologie animale
ROBERT Daniel	: Physique
ROBERT Guy	: Electrochimie
ROBERT Jacques	: Mathématiques I
THEOBALD J. Gérard	: Spectroscopie hertzienne
THIEBAUT Jean	: Pétrogr. minéralogie
TREHEL Michel	: Informatique
VERNEAUX Jean	: Zoologie-Embryologie
VIENOT J. Charles	: Physique Gle optique
WEIL Michel	: Mathématiques

Responsable des services administratifs :

M. Jean-Marie RUNGE

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ
وَقُلْ رَّبِّيَّ زِدْنِي عِلْمًا

- قرآن کریم -

"Mon Seigneur augmente ma Science"

Coran

A MON PERE,

A MA MERE,

A MA FEMME MARTINE,

A MES ENFANTS,

A MON FRERE RIBHI,

A MES SOEURS,

A MES BEAUX-PARENTS,

ET A TOUS MES AMIS.

Je suis très reconnaissant envers Monsieur MIELLOU, Professeur et responsable du service d'Analyse Numérique au L.A.N.I. (Université de BESANCON), à qui je dois mes notions en analyse numérique et qui m'a fait l'honneur d'être le Président du jury.

Je tiens à remercier :

Monsieur GOIFFON, Sous-Directeur de la Société Peugeot-Cycles,

Monsieur LEMAIRE, Professeur et Directeur du Laboratoire de Recherche Opérationnelle au C.N.A.M. à PARIS,

qui m'ont accueilli et qui me font l'honneur de participer au jury.

Je remercie bien sincèrement Monsieur ROUBELLAT, Responsable de la division systèmes complexes du L.A.A.S. de TOULOUSE qui m'a fait l'honneur de participer au jury.

Je suis très reconnaissant envers Monsieur TREHEL, Professeur et responsable du service d'Informatique au L.A.N.I. (Université de BESANCON), à qui je dois mes notions en informatique ; qui m'a fait connaître un domaine de recherche passionnant, m'a guidé de ses conseils judicieux et encouragé. Il anime une équipe dans laquelle chacun apprécie de pouvoir travailler et échanger des idées. Qu'il veuille bien trouver ici l'expression de ma profonde gratitude.

Il m'est agréable de remercier tous les membres du L.A.N.I. pour leur sympathie et le personnel du Centre de Calcul pour sa gentillesse. Je remercie tout particulièrement Madame FORSTER qui s'est acquittée avec beaucoup de patience et d'habileté de la lourde tâche dactylographique, ainsi que Monsieur LIEVREMONT qui imprime ce document.

TABLE DES MATIERES

Résumé	1
<u>INTRODUCTION</u>	2
I / RAPPEL	2
II/ POSITION DU PROBLEME	3
A) Les données	3
B) Les contraintes	4
C) Les critères	5
D) Notation	
$n \mid m \mid n_1, m_1 \mid d1, d2 \mid c$	7
<u>CHAPITRE I</u>	
Dépendance entre le 1er et 2ème traitement	9
I / DEFINITIONS ET PRINCIPES	9
1.1. Principe	9
1.1. Définition	9
1.2. Principe	10
1.2. Définition	10
II/ ORGANISATION DU TRAVAIL SUR UNE CHAINE DE 2ème TRAITEMENT	10
2.1. Résultat	11

III/ RELATION DE DEPENDANCE ENTRE LES DEUX TRAITEMENTS ..12

A) Ordonnancement de 2ème traitement fixé	12
1.3.2. Système	15
1.3.3. Système	16
B) Solution du système 1.3.3.....	16
1.3.4. Système	16
1.3.5. Système	16
1.1. Théorème	17
1.1. Corollaire	17
C) Le 2ème traitement en type est effectué sur plusieurs chaines.	
1.2. Corollaire	18
Application	18

CHAPITRE II

Ordonnancement de 1er traitement	19
	19
I / Nombre des parts n est: $1 \leq n \leq 2$	
A) $1 \mid m \mid 1, 1 \mid D1, (D2, T2) \mid C1.1 \text{ et } C2$	19
2.1. Algorithme	19
B) $2 \mid m \mid 1, 1 \mid D1, (D2, T2) \mid C1.1 \text{ et } C2$	20
2.1. Remarque	21
2.2. Remarque	21

B.1)	Système (2.1.1.)	21
B.2)	2.1.2 Système	21
	2.1.3 Système	22
	2.1. Théorème	22
	Application	23
<u>EXEMPLES NUMERIQUES -</u>		23
	1. Exemple	23
	2. Exemple	24
	3. Exemple	24
II / LE NOMBRE DE PARTS EST $n > 2$, n fixé		24
A)	Rappel	25
2.3.	Remarque	25
2.4.	Remarque	25
B)	Système d'ordre S pour l'ensemble A	
	($0 \leq S \leq p-1$)	25
2.2.1.	Définition	26
	a - système d'ordre o	26
	b - ordonnancement associé au système	
	d'ordre o	27
2.2.1.	Lemme	27
2.2.2.	Définition	
	a - système d'ordre 1 (2.2.2)	28
	b - ordonnancement de (2.2.2)	29
2.2.2.	Lemme	30

2.2.3. Définition	30
a - système d'ordre $S, (1 \leq S \leq p-1)$	30
2.2.3. Système	31
b - ordonnancement de (2.2.3)	32
2.2. Théorème	32
 c) Calcul numérique de $d_s, 0 \leq S \leq p-1$	33
 <u>2.1. Conséquence.</u>	
 III/ SOLUTION DU PROBLEME $n \mid m \mid 1,1 \mid D1,(D2,T2) \mid (C1.1)$ et C2	36
 2.3.1. Algorithme	36
2.3.1. Conséquence	43
Programmation structurée	46
2.3.2. Algorithme	46
 IV / APPLICATION	47
 Exemple numérique	47
1. Exemple	47
2. Exemple	48
3. Exemple	49
4. Exemple	51
5. Exemple	51
6. Exemple	55
 V / GENERALISATION	55
 Solution du problème $n \mid m \mid 1,m_1 \mid D1,(D2,T2,T2.1) \mid$ C1.1 et C2	55

2.5.1. Algorithme	56
 <u>CHAPITRE III</u>	
Ordonnancement du 2ème traitement	57
I / RAPPEL : Contraintes et définitions	57
3.1. Remarque	58
3.1. Définition	58
3.2. Définition	59
II / DEPENDANCE ENTRE LE 1er TRAITEMENT ET LE 2ème TRAITEMENT	59
3.1. Système	60
3.1. Conséquence	61
A) Réécriture du système (3.1)	61
(3.2) système	61
B) Equivalence entre les systèmes (3.2) et (3.1)..	61
C) Solution du système (3.2)	63
3.1. Théorème	63
III/ DETERMINATION DU NOMBRE DE JOURS DE DECALAGE ENTRE LES DEUX TRAITEMENTS	63
A) Notation	63
B) 1er cas $L_{\gamma(1)} = p$ (ou $p-1$)	64

3.3. Système	64
c) 2ème cas $L_{\gamma(1)} < p-1$	65
3.4. Système	66
D) Solution du système (3.4)	66
3.2. Théorème	68
3.** Système	68

IV / ORDONNANCEMENT MINIMUM EN TEMPS DE TRANSITION

1) L'ordonnancement commence par T_1 ou T_m	68
2) L'ordonnancement commence par T_i avec $i \neq 1$ et $i \neq m$	
Ordonnancement (a)	69
Ordonnancement (b)	69
3) Réécriture de l'ordonnancement (a) et (b)	71
3.5. Système	72

V / ORDONNANCEMENT MINIMUM DU 2ème TRAITEMENT

A) $m = 1$ (m est le nombre de type)	73
B) $m = 2$	73
<u>1er cas $L_1 \neq L_2$</u>	74
3.4.1. Système	74

2ème cas $L_1 = L_2$	75
3.3.1. Système	75
3.5.1. Résultat	75
C) Le nombre de type $m = 3$	76
3.5.1. Théorème	76
3.5.1. Algorithme	76
EXEMPLE NUMERIQUE	77
D) Le nombre de type est $m \geq 4$ (m fixé)	79
1) $L_{\gamma(1)} = L_j \geq p-1$	79
3.5.2 Théorème	80
2) $L_{\gamma(1)} = L_j < p-1$	80
3.5.1. Définition	81
Réduction du nombre de jours de décalage	81
Etape Réduction	84
3.5.2. Algorithme	85

VI/EXEMPLES NUMERIQUES

1. Exemple	86
2. Exemple	92
3. Exemple	95

VII/ GENERALISATION

Solution du problème :

$n \mid m \mid 1, m_1 \mid (D1, T1), (D2, T2) \mid C1.2 \text{ et } C2 \dots\dots\dots 99$

3.7.1. Conséquence 100

3.7.1. Algorithme 100

CHAPITRE IV

Comparaison avec Machine Flow-Shop problème 102

I / SOLUTION DU PROBLEME $1 \mid m \mid 1, m_1 \mid D1, (D2, T2.1) \mid C1 \text{ et } C2 \dots\dots\dots 102$

A) Solution du problème $1 \mid m \mid 1, 1 \mid D1, D2 \mid C1. C2 \dots\dots 102$

4.1. Algorithme 103

4.1. Théorème 103

B) Solution du problème $1 \mid m \mid 1, m_1 \mid D1, (D2, T2.1) \mid C1 \text{ et } C2 \dots\dots\dots 105$

C) Exemple numérique 106

Méthode de Johnson 106

II / SOLUTION DU PROBLEME $n \mid 1 \mid 1, m_1 \mid D1, (D2, T2.1) \mid C1 \text{ et } C2 \dots\dots\dots 109$

A) Solution du problème $n \mid 1 \mid 1, 1 \mid D1, D2 \mid C1 \text{ et } C2 \dots\dots 109$

4.2. Algorithme	110
4.2. Théorème	110
B) Exemple numérique	110
C) Solution du problème $n 1 1, m_1 D1, (D2, T2.1) C1$ et C2	112
III/ PROBLEME n x m Flow-Shop Machine	113
A) Position du problème	113
B) Heuristique	
4.3. Algorithme	115
Calcul de décalage	116
C) Application	117
1. Exemple	117
2. Exemple	121
3. Exemple	123
 <u>CHAPITRE V</u>	
Décalage fixé	124
I / MARCHE EN ARRIERE	124
A) Visualisation des contraintes	124
B) Principe de l'algorithme	127

C) Détail de l'algorithme	128
D) Application numérique	131
II / MARCHE EN AVANT	
A) Décomposition de la matrice représentant l'ordonnancement du 2ème traitement	132
B) Borne supérieure du nombre de changement	134
C) Algorithme Marche en avant	135
5.1. Algorithme	138
III/ APPLICATION NUMERIQUE	139
IV/ PROBLEME $n m 1,m_1 (D1,T3),D2 C1$	150
ANNEXE	
GESTION DE L'ORDONNANCEMENT	151
I / INTRODUCTION	152
II / LIEUX CONCERNES	154
III/ ORGANISATION CENTRALISEE	154
A) Incidents	154
B) Caractéristiques d'un ensemble d'objets donnés.	155

C) Dispositif à mettre en place	156
D) Programme de rattrapage	157
<u>D.1.</u> Programme de rattrapage pour incident avant la période	157
<u>D.2.</u> Programme de rattrapage pour incident en cours de la période	157
IV / ORGANISATION REPARTIE	160
CONCLUSION	165
BIBLIOGRAPHIE	166

R E S U M E

Il s'agit d'ordonnancer la production d'un ensemble E d'objets durant une période fixée, sur deux traitements distincts, sachant que chaque objet de E doit passer successivement par le premier puis par le deuxième traitement.

L'ensemble E admet deux partitions indépendantes :

- 1/ E peut être divisé en parts,
- 2/ E peut être divisé en types.

Des coûts de transition font qu'il est optimum de réaliser le premier traitement en parts et le second traitement par types.

On souhaite déterminer le programme du lancement le plus économique sur l'ensemble des deux traitements.

INTRODUCTION

I/ RAPPEL -

Situons le problème d'ordonnancement qui nous préoccupe par rapport aux problèmes connus. La formalisation la plus générale semble être : "Etant données n pièces qui doivent passer sur m machines dans un ordre donné et sous certaines contraintes, quel est, relativement à un critère donné, l'ordre optimal dans lequel chaque machine traite l'ensemble des pièces".

Ce problème est parfaitement défini lorsque sont précisées les données, les contraintes qui entrent en jeu et le critère à optimiser, CONWAY, MAXWELL et MILLER [5] ont proposé de caractériser ce genre de problème par une notation compacte du type :

$n|m|G|D|C$ dans laquelle :

- n et m désignent respectivement le nombre de pièces et de machines,
- G caractérise les gammes de fabrication de chacune des pièces,
- D définit les contraintes,
- C désigne le critère utilisé.

Le lecteur trouvera des détails sur G , D et C dans la thèse de B. JULLIEN [6] ; dans la formalisation du problème précédent si on substitue la tâche à la pièce et processeur à la machine, on peut utiliser le schéma :

$n|m|\gamma,\Gamma|C$ conçu par A. RINNOOY KAN [7] et J. LENSTRA [8] dans lequel :

- γ représente les contraintes de séquençement,
- Γ représente les conditions restrictives sur les tâches.

Le lecteur trouvera des détails sur γ et Γ dans la thèse de

J. LABETOULLE [9]. Nous allons utiliser le principe de ce type de notations compactes en l'adaptant à notre problème d'ordonnancement.

II/ POSITION DU PROBLEME -

Les problèmes d'ordonnancement que nous traitons sont caractérisés par un ensemble de données. Pour qu'ils soient complètement déterminés, il faut encore en préciser les contraintes et les critères.

A/ Les données -

Soit E un ensemble d'objets (pièces).

- 1ère partition -

. le nombre n est le nombre des parts $p_1, p_2, \dots, p_n$,

on note par $|p_i| = \text{card } p_i, \quad |p_i| \geq 1$.

- 2ème partition -

. le nombre m est le nombre des types $T_1, T_2, \dots, T_m$,

on note par $|T_j| = \text{card } T_j, \quad |T_j| \geq 1$.

- Pour le 1er traitement -

. la durée d'exécution d'un objet de la part p_i est b_i .

. le temps de transition, si on passe d'un objet de la part p_i à un objet de la part p_k , est c_{ik} .

- Pour le 2ème traitement -

. la durée d'exécution d'un objet du type T_j est d_j ,

. le temps de transition, si on passe d'un objet du type T_j à un objet de type T_k , est e_{jk} .

. soit M la matrice des besoins, $M = (a_{ji}) \quad 1 \leq i \leq n, 1 \leq j \leq m$ ou a_{ji} est le nombre d'objets du type T_j en part P_i . Alors le nombre d'objets à produire durant la période est :

$$\text{card } E = \sum_{i=1}^n \sum_{j=1}^m a_{ji}, \quad a_{ji} \in \mathbb{N}$$

- (02) - Toutes les parts sont disponibles au début de la période de travail de premier traitement
- (M2) - Toutes les chaînes de 2ème traitement sont disponibles après un nombre de jours de décalage par rapport à la chaîne de 1er traitement
- (M3) - Toutes les chaînes de 1er et 2ème traitement sont dans un seul état : travail sur les objets en part (respectivement en type) pour la chaîne de 1er traitement (resp. les chaînes de 2ème traitement)

Remarque : dans le cas des problèmes définis dans I, les machines peuvent être en attente ce qui est interdit ici.

- (03) - Toutes les parts sont différentes et importantes
- (04) - Tous les types sont différents et importants
- (M4) - Tout programme (ordonnancement) doit respecter l'engagement journalier de chaque chaîne
- (OM1) - Chaque objet de E doit passer successivement dans la chaîne de 1er traitement et une seule chaîne de 2ème traitement
- (OM2) - Un objet qui passe par le 2ème traitement doit être passé par le 1er traitement un des jours précédents (donc pas le jour même)
- (OM3) - Chaque chaîne traite plusieurs objets à la fois
- (OM4) - Pour chaque chaîne de 1er et 2ème traitement toute tâche commencée doit aller jusqu'à son complet achèvement
- (05) - Chaque objet de E est dans l'un des trois états suivants : en attente ou bien en traitement ou bien terminé

C/ Les critères -

Rappelons brièvement que les critères des problèmes d'ordonnement "d'atelier" sont basés sur :

les dates d'achèvement et les temps de passage, les retards, les en-cours et le coefficient d'utilisation et les divers de réemploi.

Pour les problèmes d'ordonnement :

- "job-shop" et "flow shop" (par exemple [3], [4], [15], [14], [13])
- à contraintes disjonctives [12]
- à ressources variables [10]
- dans la fabrication en série [11] ,

Pour notre problème on préfère parler de critères basés sur : le nombre de jours de décalage entre les deux traitements ; le temps de transition d'une part à une autre ou d'un type à un autre type ; pour nous le retard n'est pas admis, au lieu de dire des durées de réemploi on préfère parler du changement de rythme d'une chaîne, on ne cherchera pas à optimiser la durée totale d'exécution car la période p est fixée d'avance, par contre on cherchera à minimiser le temps de transition pour un traitement.

Pour une période donnée, pour le 1er traitement, considérons qu'on va passer des objets d'une certaine part, soit p_{i_1} , puis d'une autre part $p_{i_2}, \dots$, puis $p_{i_k}, \dots$, puis p_{i_r} (on peut être amené au cours de la période à prendre des objets non encore pris d'une part déjà prise, donc certains indices peuvent être égaux, mais les indices de deux p_{i_k} successifs sont différents). De même pour le second traitement on a la suite $T_{j_1}, \dots, T_{j_k}, \dots, T_{j_s}$.

Alors le temps pour le 1er traitement est :

$$\sum_{i=1}^n \left(\sum_{j=1}^m a_{ji} b_i \right) + \sum_{k=1}^{r-1} c_{i_k, i_{k+1}} \dots \quad r \geq n ;$$

et le temps pour le 2ème traitement est :

$$\sum_{j=1}^m \left(\sum_{i=1}^n a_{ji} d_j \right) + \sum_{k=1}^{s-1} e_{j_k, j_{k+1}} \dots \quad s \geq m ;$$

S'il y a plusieurs chaînes, on somme sur l'ensemble des chaînes. Il y a clairement dans la somme une partie fixe : c'est la partie de traitement proprement dit :

$$\sum_{i=1}^n \sum_{j=1}^m a_{ji} (b_i + d_j)$$

la partie "temps de transition" :

$$(C \ 1) \quad \sum_{k=1}^{r-1} c_{i_k, i_{k+1}} + \sum_{k=1}^{s-1} e_{j_k, j_{k+1}}$$

dépend de l'ordonnancement.

On note par (C 1.1.) le critère qui consiste à minimiser la partie temps de transition de 1er traitement :

$$\sum_{k=1}^{r-1} c_{i_k, i_{k+1}}$$

On note par (C 1.2.) le critère qui consiste à minimiser la partie temps de transition de 2ème traitement : $\sum_{k=1}^{s-1} c_{j_k, j_{k+1}}$

La contrainte (M2) indique que la chaîne de 1er traitement prend un nombre de jours d'avance par rapport aux chaînes de 2ème traitement, on note par (C 2) le critère qui consiste à minimiser le nombre de jours de décalage entre les deux traitements.

D/ Résumé et notations -

Dans les chapîtres suivants nous allons tenter de résoudre quelques problèmes concrets et théoriques, nous utiliserons la notation suivante :

$$n|m|n_1, m_1|D1, D2|C$$

où :

- n et m désignent respectivement le nombre des parts et des types,
- n_1 et m_1 désignent respectivement le nombre des chaînes de 1er traitement et des chaînes de 2ème traitement,
- D1 et D2 les contraintes pour le 1er et le 2ème traitement,
- C le critère à optimiser.

Pour résoudre ces problèmes, nous avons opéré ainsi.

D'abord nous avons étudié le cas où chaque traitement admet une seule chaîne, ensuite nous avons généralisé à plusieurs chaînes de 2ème traitement.

En effet, dans le Chapitre I, on fixe l'ordonnancement de 2ème traitement, on détermine la relation de dépendance entre les deux traitements, cette relation respecte toutes les contraintes et donne le nombre de jours de décalage minimum.

Le Chapitre II, avec les mêmes hypothèses que le chapitre I, nous permet de généraliser la relation ci-dessus, mais en cherchant à définir l'ordonnancement minimum de 1er traitement avec un décalage minimum.

Dans le Chapitre III, on fixe l'ordonnancement de 1er traitement, on définit une nouvelle relation de dépendance, puis on généralise cette relation pour obtenir l'ordonnancement de coût minimum en temps pour le 2ème traitement.

Le Chapitre IV aborde le problème où aucun ordonnancement n'est fixé, on donne des solutions dans le cas (une part, m types), puis (n parts, un type), et ceci nous conduit à une comparaison avec le problème flow-shop machines.

Dans le Chapitre V, on détermine l'ordonnancement minimum de 1er traitement quand le décalage est fixé entre les deux traitements ; puis enfin, on termine avec une réflexion sur la gestion de l'ordonnancement (voir annexe).

CHAPITRE I

DEPENDANCE ENTRE LE 1er ET 2ème TRAITEMENT

Résumé -

Ce chapitre est destiné à présenter les outils mathématiques nécessaires à la suite, on donnera l'organisation de travail sur une chaîne de 2ème traitement.

I/ DEFINITIONS ET PRINCIPES -

Principe 1.1. -

Minimiser le temps de transition pour le 1er traitement (C 1.1) revient à travailler par part (l'ordre à définir), c'est-à-dire si on traite un objet d'une part, on continue à traiter sans interruption tous les objets de cette part.

Définition 1.1. -

a) Si le nombre des chaînes de 1er traitement est ($n_1=1$) ; alors la capacité moyenne journalière S est : le quotient de temps traitement "proprement dit" pour le 1er traitement par le nombre de jours de la période :

$$S = \sum_{i=1}^n \sum_{j=1}^m a_{ji} b_i : P \quad .$$

b) Soit $h \in \{1, \dots, p\}$;

soient $p_1, p_2, \dots, p_l$ les parts engagées le $h^{i\text{ème}}$ jour de 1er traitement, (engagées dans cet ordre). Alors la durée de travail effective journalière est :

$$S_h = S + \sum_{i=1}^{(l-1)} c_{i,i+1} .$$

Principe 1.2. -

Minimiser le temps de transition pour une chaîne de 2ème traitement revient à travailler par type, (l'ordre à définir), c'est-à-dire si on commence à traiter un objet du type T_j , on continue à traiter les objets du type T_j sans interruption jusqu'au dernier.

Définition 1.2. -

a) Si le nombre des chaînes de 2ème traitement est ($m_1=1$) ; alors la capacité moyenne journalière S' est le quotient de temps traitement "proprement dit" de 2ème traitement, par le nombre de jours de la période P

$$S' = \frac{\sum_{j=1}^m \sum_{i=1}^n a_{ji} d_j}{P} :$$

b) Soit $h \in \{1, \dots, p\}$;

soient $T_1, T_2, \dots, T_k$ les types engagés (pris dans l'ordre) le $h^{i\text{ème}}$ jour de 2ème traitement. Alors la durée de travail journalière effective est :

$$S'_h = S' + \sum_{j=1}^{(k-1)} e_{j,j+1} .$$

II/ ORGANISATION DU TRAVAIL SUR UNE CHAÎNE DE 2ème TRAITEMENT -

Dans ce paragraphe on rappelle l'organisation détaillée par M. TREHEL [1]. Le travail consiste à attacher aux objets divers accessoires, pour un type d'objets donné, ce sont les mêmes accessoires.

De plus d_j dépend de T_j ; la pose d'accessoires se fait à la chaîne, considérons f employés sur la chaîne, quand on passe d'un type où le temps de travail par employé est t_1 à un type où le temps de travail est t_2 avec $t_1 < t_2$ ($t_j = d_j / f$), on a cette situation :

1er objet du type 2	dernier objet du type 1	objet du type 1		Objet du type 1
---------------------------	-------------------------------	-----------------------	-------	-----------------------

1er
employé
de la chaîne

f^{ième} employé
de la chaîne
(le dernier)

Quand arrive le 1er objet du type 1, il faut ralentir le rythme pour que le 1er employé de la chaîne ait le temps de poser l'accessoire qu'il doit poser. Donc (f-1) employés disposent du temps inutile ($t_2 - t_1$).

Quand arrive le 2ème objet du type 2, les 2 premiers employés auront le bon rythme (t_2) ; les (f-2) autres disposent du temps inutile ($t_2 - t_1$).

Ainsi e_{12} , temps de transition est $\frac{f(f-1)}{2} (t_2 - t_1)$.

Si $t_1 > t_2$ il y a accélération de rythme dès que le dernier employé de la chaîne attaque le 1er objet du type 2, le temps e_{12} est encore $\frac{f(f-1)}{2} (t_1 - t_2)$. La matrice de transition est donc symétrique.

Il est clair que l'ordonnancement optimum pour le traitement "montage d'accessoires" consiste à travailler par type, dans l'ordre croissant des t_j , ou l'ordre décroissant des t_j .

Résultat 2.1. -

Avec l'organisation du travail ci-dessus, et en respectant le principe 1.2., en prenant les types dans l'ordre croissant ou décroissant de d_j , le temps de transition minimum pour le 2ème traitement ($m_1=1$) est :

$$\sum_{k=1}^{s-1} e_{k,k+1} = \sum_{j=1}^{m-1} |d_j - d_{j+1}| = d_1 - d_m$$

en supposant $d_1 \geq d_2 \geq \dots \geq d_m$.

Avant de commencer le paragraphe III, donnons un exemple d'ordre de grandeur des variables pour le 2ème traitement, puis pour le 1er traitement.

card E $\geq$ 5000

2ème traitement

m = 50 types
 d_j = 20 à 60 unités de temps
 t_j = 1 à 3 unités de temps
 f = 20 employés

1er traitement

n = 15 parts
 b_i = 1/2 unité de temps (par exemple)
 c_{ik} = 15 unités de temps (par exemple)

III/ RELATION DE DEPENDANCE ENTRE LES DEUX TRAITEMENTS -

A/ Ordonnancement de 2ème traitement fixé -

Considérons d'abord que chaque traitement est constitué d'une seule chaîne, avec la nouvelle contrainte suivante :

(T2) L'ordonnancement en type de 2ème traitement est fixé à l'avance.

Nous allons mettre en évidence un système qui respectera les contraintes : (T2) ; (M3) toutes les chaînes en état de travail ;

(M4) l'engagement journalier de chaque chaîne ;

(OM1 et OM2) un objet qui passe par le 2ème traitement doit être passé par le 1er traitement l'un des jours précédents, ceci implique que la date limite de son passage par le 1er traitement est au plus la veille de son passage par le 2ème, et enfin, (M2) la chaîne de 2ème traitement est disponible après un nombre de jours de décalage par rapport à la chaîne de 1er traitement.

Soit d le nombre de jours de décalage entre les deux traitements.

Comme l'ordonnancement en type de 2ème traitement est fixé, à l'aide de la matrice des besoins $M = (a_{ji})$, on peut représenter l'ordonnancement de 2ème traitement, par la matrice X suivante, en fonction des parts.

MATRICE X -

	p_1	p_2	p_3	p_n
1er jour de 2ème traitement	x_{11}	x_{12}	x_{13}	x_{1n}
2ème jour de 2ème traitement	x_{21}	x_{22}	x_{23}	x_{2n}
$k^{ième}$ jour de 2ème traitement	x_{k1}	x_{k2}	x_{k3}	x_{kn}
$p^{ième}$ jour de 2ème traitement	x_{p1}	x_{p2}	x_{p3}	x_{pn}

où x_{ki} = le nombre des objets de différent type en part p_i , engagés le $k^{ième}$ jour dans le 2ème traitement,

On appelle $Q'_k = \sum_{i=1}^n x_{ki}$: le nombre d'objets engagés le $k^{ième}$ jour dans le 2ème traitement,

Les Q'_k objets : leur temps de travail dans le 1er traitement est noté :

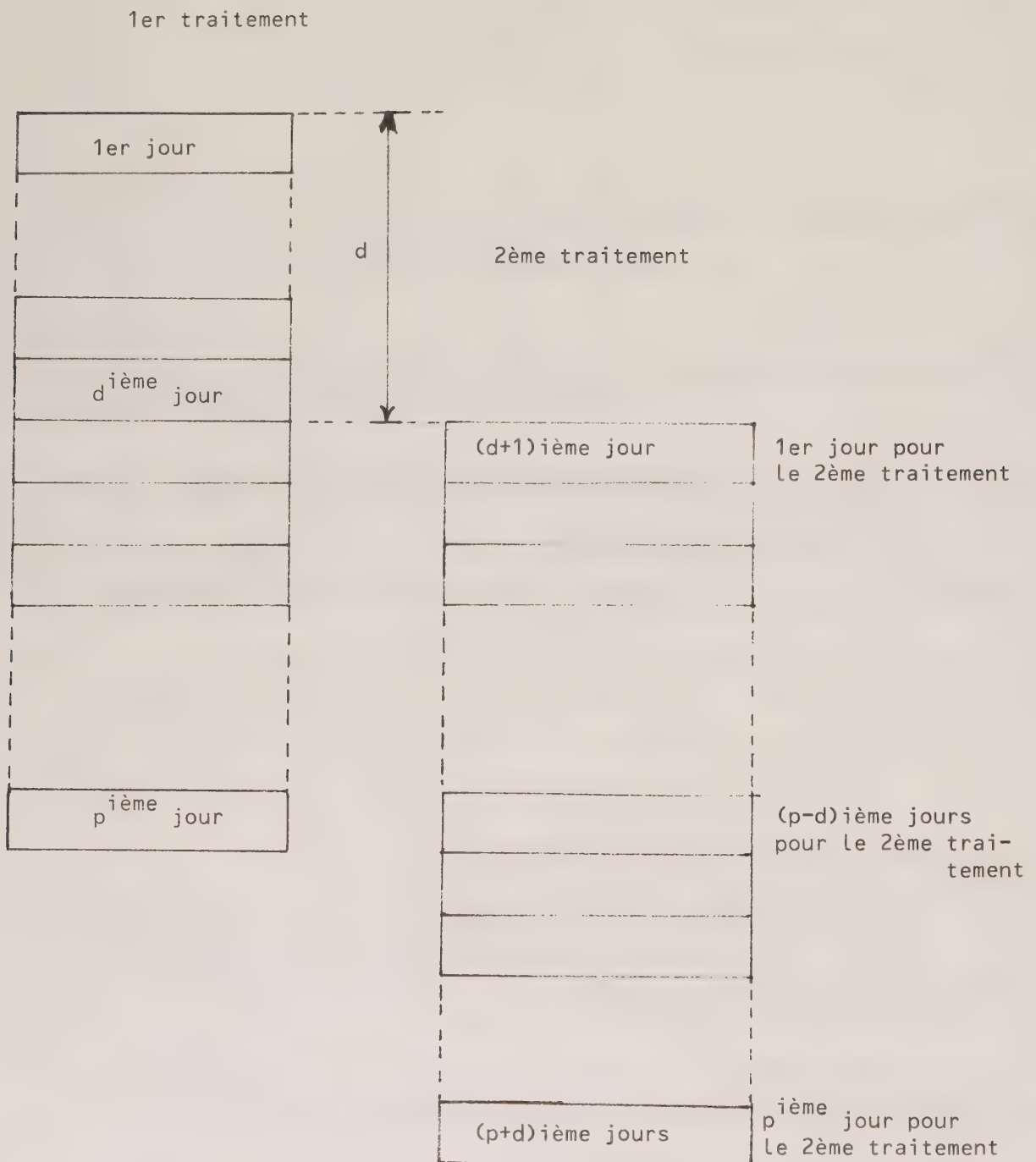
$$Q_k = \sum_{i=1}^n x_{ki} \cdot b_i$$

pour simplifier les notations nous considérons que :

(3.1) L'engagement journalier pour le 2ème traitement est "représenté" par Q_j . Par conséquent l'engagement journalier moyen pour le 1er traitement S est égal à : $\sum_{k=1}^p Q_k : P$

Donnons maintenant le schéma représentant les chaînes de 1er et 2ème traitement avec un nombre de jours de décalage d (figure 1.2).

(Figure 1.2.)



Un tel d doit vérifier :

pour le 1er jour de 2ème traitement $Q_1 \leq d \times s$

(d premiers jours de 1er traitement)

Pour le 2ème jour de 2ème traitement $Q_1 + Q_2 \leq (d+1) \times s$

($(d+1)$ premiers jours de 1er traitement)

⋮

Pour le $k^{\text{ième}}$ jour de 2ème traitement $Q_1 + Q_2 + \dots + Q_k \leq (d+k-1) \times s$

Il est clair que Q_k est fait au moins à la date $(k-1)^{\text{ième}}$ jour de 2ème traitement (regarder figure 1.2).

Pour le $(p-d)^{\text{ième}}$ jour de 2ème traitement :

$$\sum_{k=1}^{(p-d)} Q_k \leq (d+(p-d)-1)s = (p-1) \times s$$

On rappelle que le $(p-d)^{\text{ième}}$ jour de 2ème traitement correspond au dernier jour (le $p^{\text{ième}}$) de travail pour le 1er traitement, en effet pour le $(p-d+1)^{\text{ième}}$ jour de 2ème traitement, nous avons :

$$\sum_{k=1}^{(p-d+1)} Q_k \leq (d+(p-d+1)-1) = p \times s$$

Or pour tout $k \in \{p-d+1, p-d+2, \dots, p\}$ nous avons :

$$\sum_{j=1}^k Q_j \leq \sum_{j=1}^p Q_j = p \times s = (d+(p-d+1)-1) \times s < (d+k-1) \times s$$

On en déduit que d le nombre de jour de décalage pour le 1er traitement par rapport au 2ième traitement doit vérifier :

$\forall k \in \{1, \dots, p-d\}$; $k^{\text{ième}}$ jour de 2ème traitement :

$$\sum_{j=1}^k Q_j \leq (d+k-1) \times s \quad \dots \quad 1.3.2$$

Soit $\mathcal{D} = \{d; d \in \mathbb{N}^*; \text{ et } d \text{ vérifie (1.3.2)}\}$

$\mathcal{D} \neq \emptyset$, il suffit de prendre $d=p$ solution évidente.

Conséquence -

Soit $d_0 = \min_{d \in \mathbb{D}} \{d\}$, alors d_0 est l'entier minimum vérifiant
(1.3.2.).

Soit le système suivant avec d_0 ,

$\forall k \in \{1, \dots, p-d_0\}$

$$\sum_{j=1}^k q_j \leq (d_0 + k - 1) \cdot s \quad \dots\dots (1.3.3)$$

B/ Solution du système (1.3.3) -

Soit $d_k \in \mathbb{N}^*$, l'entier minimum vérifiant pour tout $k \in \{1, \dots, p-1\}$

$$\sum_{j=1}^k q_j \leq (d_k + k - 1) \cdot s \quad \dots\dots (1.3.4)$$

il suffit de prendre $d_k = \left\lceil \sum_{j=1}^k q_j : s \right\rceil + 1 - k$;

Soit $k_0 \in \{1, \dots, p-1\}$, avec $d_{k_0} = \max_{k=1}^{p-1} \{d_k\}$;

Il est clair que : $\forall k \in \{1, \dots, p-1\}$

$$\sum_{j=1}^k q_j \leq (d_k + k - 1) \cdot s \leq (d_{k_0} + k - 1) \cdot s$$

En particulier $\forall k \in \{1, \dots, p-d_{k_0}\}$

$$\sum_{j=1}^k q_j \leq (d_{k_0} + k - 1) \cdot s \quad \dots\dots (1.3.5)$$

On en déduit que d_{k_0} est une solution de (1.3.2) ; $d_{k_0} \in \mathbb{D}$.

Donc $d_{k_0} \geq d_0$.

Théorème 1.1. -

L'entier minimum solution du système (1.3.3) est d défini par :

$$d = \max_{k=1}^{p-1} \left\{ \left\lceil \sum_{j=1}^k Q_j : s \right\rceil + 1 - k \right\}$$

Démonstration -

On vient de voir que $d = d_{k_0} \geq d_0$; montrons que l'inégalité ne peut pas être stricte.

Supposons que $d > d_0$. Alors $-d < -d_0$ ce qui entraîne $p-d < p-d_0$; en particulier $\{1, \dots, p-d\} \subset \{1, \dots, p-d_0\}$.

Or d'après (1.3.5) d est l'entier minimum vérifiant pour $k \in \{1, \dots, p-d\}$ $\sum_{j=1}^k Q_j \leq (d+k-1) \times s$.

Or d'après (1.3.3) nous avons pour $k \in \{1, \dots, p-d_0\} \supset \{1, \dots, p-d\}$

$$\sum_{j=1}^k Q_j \leq (d_0 + k - 1) \times s \quad \text{donc } d_0 \text{ est un majorant de } d, d_0 \geq d,$$

or nous avons supposé $d > d_0$, absurde et $d = d_0$.

Corollaire 1.1. -

Si pour $k=1, \dots, p-1$; avec d_k solution de (1.3.4),

si $d_1 \geq d_2 \geq \dots \geq d_{p-1}$, alors d'après le théorème (1.1)

d_1 est la solution de (1.3.3).

C/ Le 2ème traitement en types est effectué sur plusieurs chaînes -

On considère que toutes les contraintes de ce paragraphe III.A/ sont valables, de plus la donnée m_1 n'est plus égale à 1, mais $m_1 > 1$, m_1 fixé dans $\mathbb{N}^*$, on ajoute la contrainte (T 2.1) :

- Les m_1 chaînes de 2ème traitement travaillent en parallèle, c'est-à-dire commencent le travail à la même date pour la période p .

Avec les mêmes notations de ce paragraphe : si $x_1, x_2, \dots, x_{m_1}$

représentent l'engagement journalier des chaînes de 2ème traitement.

Corollaire 1.2. -

La solution du problème $n|m|1, m_1|d_1; (D2; T2; T2.1)|C2$

- qui consiste à trouver le nombre de jours de décalage minimum entre les deux traitements- est donnée par le théorème (1.1) en prenant

$$X = X_1 + X_2 + \dots + X_{m_1}.$$

Application -

On considère 3 chaînes de montage d'accessoires principales travaillant en parallèle, pour le 2ème traitement, la répartition entre ces 3 chaînes - {organisation du travail II ; (et des petits choix annexes)} - obéit à une règle liée à la qualification des employés des chaînes, un type donné passe préférentiellement sur une chaîne donnée, si besoin est, sur une seconde, si possible, jamais sur une 3ème ; on constate que le 1er traitement est environ 3 fois plus rapide que le 2ème traitement.

CHAPITRE II

ORDONNANCEMENT DE 1er TRAITEMENT

Résumé -

Dans ce chapitre on s'intéressera d'abord au cas d'une seule chaîne par traitement ; après on donnera la généralisation pour plusieurs chaînes de 2ème traitement. Bien entendu, le but est de déterminer l'ordonnancement minimum en travaillant par part avec un nombre de jours de décalage minimum.

I/ NOMBRE DES PARTS n EST $1 \leq n \leq 2$ -

A/ $1|m|1,1|D1,(D2,T2)|C1.1$ et $C2$ -

(1 part/m types/1 chaîne, 1 chaîne/contraintes 1er et 2ème traitement/critère à optimiser).

Rappelons que l'ordonnancement de 2ème traitement est fixé en type ; $(T2)$, de plus il faut respecter toutes les contraintes, puisque le nombre des parts est égal à 1, il est évident que le temps de transition de 1er traitement est nul ; $(C1.1)$, il reste à préciser dans ce cas l'ordre de passage des objets dans le 1er traitement, de plus à minimiser le nombre de jours de décalage d : $(C2)$.

Algorithme 2.1 -

La solution du problème $1|m|1,1|D1,(D2,T2)|C1.1$ et $C2$ est donné par le théorème (1.1) pour $(C2)$ et l'ordonnancement de 1er traitement est le même que le 2ème traitement.

Démonstration -

Avec le théorème (1.1), le nombre de jours de décalage d est minimum, de plus les objets d'un type T_j précèdent les objets d'un type T_k dans le 1er traitement, si T_j précède T_k dans le 2ème traitement.

B/ 2|m|1,1|D1,(D2.T2)|C1.1 et C2 -

Soient p_1 et p_2 les deux parts de la période, $(T2)$ l'ordonnancement de 2ème traitement est fixé, de plus on utilise la représentation définie en II. III. A/. On rappelle que la durée d'exécution d'un objet de la part p_i est b_i , de plus le temps de transition si on passe d'un objet de la part p_i à un objet de la part p_k est $c_{ik} = c_{ki} = c$, c'est une constante qui ne dépend pas de p_i .

Soit X la matrice représentant l'engagement journalier de 2ème traitement : figure (2.1).

figure(2.1)

p_1	p_2	
x_{11}	x_{12}	1er jour de 2ème traitement
x_{21}	x_{22}	2ème jour de 2ème traitement
x_{k1}	x_{k2}	k ième jour de 2ème traitement
x_{p1}	x_{p2}	p ième jour de 2ème traitement

Notation -

Soit A l'ensemble d'indice associé à p_1 et p_2 ; $A = \{1,2\}$ pour $i \in A$, on rappelle que $|p_i| = \text{card } p_i = \sum_{k=1}^p x_{ki}$. En prenant comme origine des temps le début du 2ème traitement, on note L_i le numéro du jour de 2ème traitement auquel la part p_i intervient pour la 1ère fois ; ainsi $x_{1i} = x_{2i} = \dots = x_{(L_i-1,i)} = 0$ et $x_{L_i,i} \neq 0$.

Soit une partition de A en plusieurs sous ensembles, $A_l = \{i, i \in A \text{ et } L_i = l\}$.

Remarque (2.1) -

Dans ce qui suit on s'intéresse uniquement au sous-ensemble A_l tel que $1 \leq l \leq p-1$; en effet si $A_p \neq \emptyset$, alors $\text{Card } A_p = 1$, puisque $A_1 \neq \emptyset$ et $\text{card } A = 2$. On considère la part p_i , $i \in A_p$ la dernière part pour l'ordonnancement de 1er traitement, car p_i , $i \in A_p$ n'a aucune influence sur le nombre de jours de décalage et l'engagement journalier de $p^{\text{ième}}$ jour de 2ème traitement sera respecté car $d \in \mathbb{N}^*$. Par contre, pour l'ordonnancement en parts de 1er traitement, il faudra en tenir compte pour minimiser le temps de transition dans certains cas.

Remarque (2.2) -

Nous allons définir des systèmes qui donneront le programme du lancement le plus économique pour le 1er traitement, avec un décalage minimum ; ceci dans le cas où $n = \text{nombre de parts} = 2$, la justification et la démonstration de ces systèmes est faite dans le cas général où $n > 2$.

B.1/ Soit $A_p \neq \emptyset$ -

Soit d l'entier minimum vérifiant le système suivant :

$$\forall k \in \{1, \dots, p-d\} \quad \sum_{l=1}^k x_{li} \cdot b_i \leq (d+k-1) \cdot S \quad \dots\dots (2.1.1)$$

Alors l'ordonnancement de 1er traitement est p_i , $i \in A_1$, puis p_j , $j \in A_p$.

B.2/ Soit $A_p = \emptyset$ (regarder la figure 2.2)

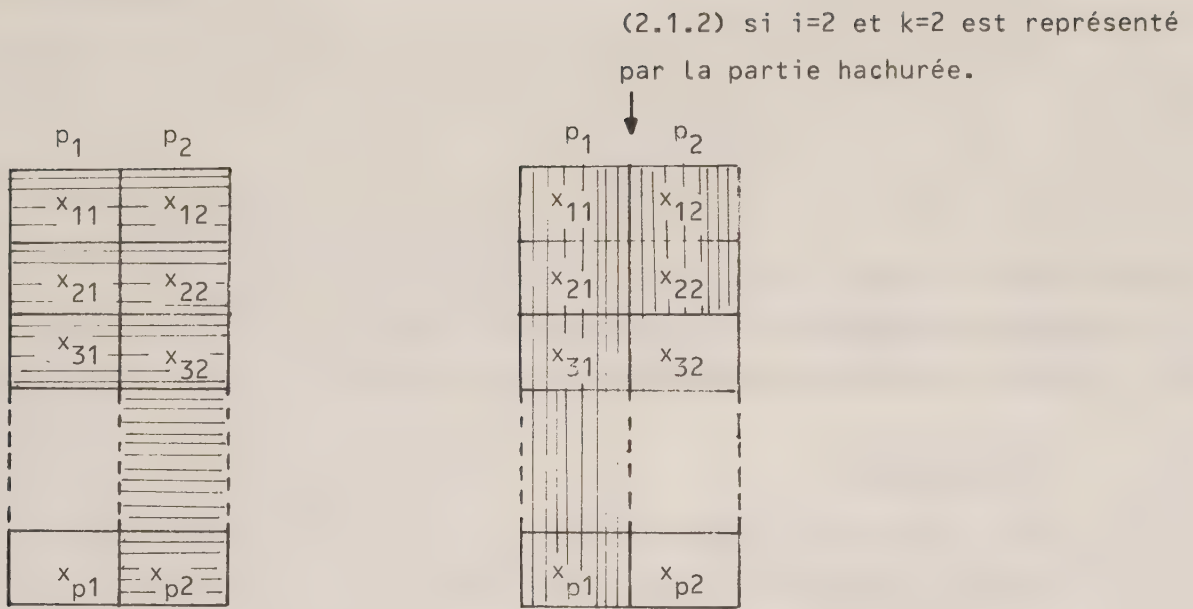
a) soit d l'entier minimum vérifiant : $\exists i \in A$ tel que :

$$\forall k \in \{1, \dots, p-d\} \quad |p_j| \cdot b_j + \sum_{l=1}^k x_{li} \cdot b_i \leq (d+k-1) \cdot s \quad \dots\dots (2.1.2)$$

Alors l'ordonnancement de 1er traitement en part est :

p_j , $j \neq i$ puis p_i .

figure (2.2)



(2.1.2) si $i=1$ est représenté par la partie hachurée pour $k=3$.

b) Supposons qu'il existe k_0 , $1 < k_0 \leq p-1$ tel que $A_{k_0} \neq \emptyset$, $A_{k_0} = \{i_{k_0}\}$; soit $A_1 = \{i_1\}$.

Soit d l'entier minimum vérifiant le système :

$$(2.1.3) \quad \begin{cases} \forall k \in \{1, \dots, k_0-1\} & \sum_{l=1}^k x_{li_1} \cdot b_{i_1} \leq (d+k-1) \cdot S \\ \forall k \in \{k_0, \dots, p-d\} & |p_{i_1}| \cdot b_{i_1} + \sum_{l=k_0}^k x_{li_{k_0}} \cdot b_{i_{k_0}} \leq (d+k-1) \cdot S \end{cases}$$

Alors l'ordonnancement en part de 1er traitement est p_{i_1} puis $p_{i_{k_0}}$.

Théorème 2.1. -

Si l'ordonnancement de 2ème traitement est fixé, alors l'ordonnancement de 1er traitement admettant le temps de transition minimum et respectant l'engagement journalier de chaque chaîne avec un nombre de jours de décalage le plus petit est déterminé par :

1. Le système (2.1.1) si $A_p \neq \emptyset$,
2. Le système (2.1.2) si $\text{card } A_1 = \text{card } A$,
3. Le système vérifié par le plus petit entier des entiers minimum vérifie (2.1.2) ou (2.1.3) si ($A_p = \emptyset$ et $\text{card } A_1 = 1$).

Démonstration -

Regarder le cas où le nombre de parts $n > 2$, ainsi pour le calcul numérique de d vérifiant chaque système, en particulier les systèmes d'ordre 1 et l'algorithme (2.3.1).

Application -

Le 1er traitement consiste à peindre les objets, les n parts correspondent à n peintures ; b_i est indépendant de p_i ; le temps de transition d'une peinture à l'autre ne dépend pas des peintures, c'est une constante c , l'ordonnancement optimum consiste à travailler par peinture ; (principe 1.1), le temps de transition est $(n-1) \times c$.

Exemples numériques -

Pour l'exemple 1, 2 et 3 on suppose que $p = 5$ jours, la durée d'exécution d'un objet est la même ($b_i = 1, i = 1, 2$) ;

Exemple 1 -

Nous sommes dans le cas 2 du théorème 2.1, car :

$\text{card } A_1 = \text{card } A = 2$ avec $|p_1| = 1500$, $|p_2| = 1000$ objets. S = engagement journalier de 1er traitement est $(|p_1| \times 1 + |p_2| \times 1) : 5 = 500$ unités de temps.

Le système (2.1.2) est vérifié pour $i = 2$ avec un nombre de décalage égal à 3 jours, on trouve pour $i=1$ le nombre de jours de décalage est de 4 jours.

	p_1	p_2	
	200	300	1
	500	0	2
	300	300	3
	250	250	4
	200	200	5

Par conséquent, l'ordonnancement optimum : on commence à peindre les objets de part p_2 jusqu'au dernier objet, puis on termine avec les objets de la part p_1 .

Exemple 2 -

card $A_1=1$, card $A_4=1$, $|p_1|=2000$,
 $|p_2|=500$ objets, $S = 500$ unités
 de temps.

Nous sommes dans le cas (3) du
 théorème 2.2.

Le système 2.1.2 donne $d = 2$ jours
 ordonnancement commençant par
 peindre les objets de p_2 puis on
 termine avec p_1 .

p_1	p_2
500	0
500	0
500	0
250	250
250	250

Le système 2.1.3 donne $d = 2$, l'ordre des peintures est p_1
 puis p_2 . On constate que les deux ordonnancements sont valables.

Exemple 3 -

card $A_1=1$, card $A_4=1$, $|p_1|=1500$ objets
 $|p_2|=1000$ objets, $S = 500$ unités.
 Dans cet exemple, le système
 (2.1.2) est vérifié pour $d = 3$,
 ordre p_2, p_1 avec le système
 (2.1.3) le nombre de jours de
 décalage est $d = 1$ et l'ordre
 de peinture est p_1 puis p_2 .

p_1	p_2
500	0
500	0
500	0
0	500
0	500

II/ LE NOMBRE DE PARTS EST $n > 2$, n FIXE -

Ce paragraphe II présente l'outillage mathématique pour trouver
 d minimum (d = nombre de jours de décalage entre les deux traitements).
 L'ordre du 2ème traitement étant fixé. Le paragraphe III suivant utilisera
 cet outillage pour développer un algorithme.

A/ Rappel -

On utilise la représentation définie en (II, III.A) ; la matrice $X = (x_{ji})$ avec $i \in \{1, \dots, n\}$, n nombre de parts, et $j \in \{1, \dots, p\}$, p est le nombre de jours de travail de la période, x_{ji} le nombre d'objets en part p_j en différents types engagés le $j^{\text{ième}}$ jour de 2ème traitement.

En prenant comme origine du temps le début de travail de 2ème traitement, on note par L_i le jour de 2ème traitement où la part p_i intervient pour la 1ère fois, ainsi $x_{1i} = x_{2i} = \dots = x_{L_i-1,i} = 0$ et $x_{L_i,i} \neq 0$. De plus $A = \{1, \dots, n\}$ ensemble d'indices associé aux parts $p_1, \dots, p_i, \dots, p_n$ pour $i \in A$, rappelons que $|p_i| = \text{card } p_i = \sum_{k=1}^p x_{ki}$, b_i est la durée d'exécution d'un objet de la part p_i .

Remarque 2.3. -

L'ordonnancement en part qu'on cherche à déterminer doit respecter le principe (1.1) à savoir travail par part sans interruption. De plus, le temps de transition, si on passe d'un objet d'une part p_i à un objet d'une part p_k est $C_{ik} = C$, constante ne dépendant pas des parts, donc en respectant le principe (1.1), le temps de transition de 1er traitement est minimum et égal à $\sum_{k=1}^{n-1} C_{p_k, p_{k+1}} = (n-1) \times C$, donc le critère (C.1.1) est satisfait.

Remarque 2.4. -

Soit une partition de A en plusieurs sous-ensembles $A_l = \{i, i \in A, L_i = l\}$. On s'intéresse toujours aux sous-ensembles $A_l \neq \emptyset$ pour $1 \leq l \leq p-1$; par la suite on considère toujours l'ensemble $A = A - A_p$, si $A_p \neq \emptyset$ alors les parts $p_i, i \in A_p$ sont les dernières parts pour l'ordonnancement de 1er traitement.

B/ Système d'ordre s pour l'ensemble A ($0 \leq s \leq p-1$) -

- Soit la part p_i et $i \in A$, on note par $A^i = A - \{i\}$,
- les systèmes que nous allons définir sont très utiles pour satisfaire le critère ((C1.1) et (C2)) avec les données et les contraintes définies intérieurement,
- on considère que $\sum_{\emptyset} |p_j| \cdot b_j = 0$

Définition 2.1. - (système d'ordre 0)

a - On appelle système d'ordre 0 pour l'ensemble A , le système (2.2.1) qui est vérifié par l'entier minimum $d_0 \in \mathbb{N}^*$, tel que :

$$\exists i \in A$$

$$\forall k \in \{1, \dots, p-d_0\} \quad \sum_{A^i} |p_j| \cdot b_j + \sum_{l=1}^k x_{li} \cdot b_i \leq (d_0+k-1) \cdot s$$

b - L'ordonnancement en parts pour le 1er traitement associé au système (2.2.1) est défini ainsi :

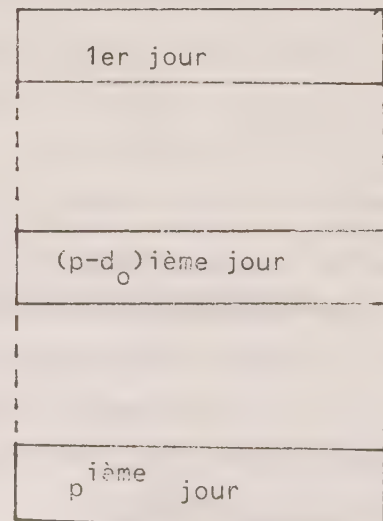
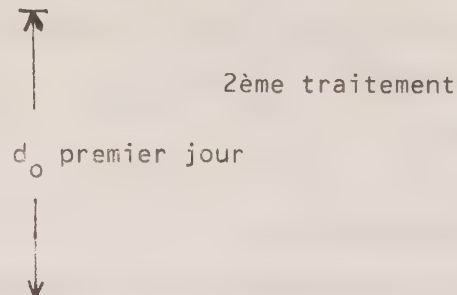
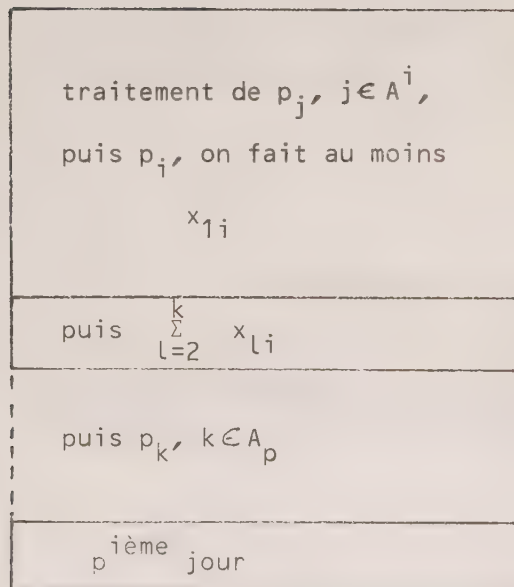
d'abord on traite les parts p_j , $j \in A^i$

puis on passe aux objets de la part p_i

et on termine par p_k , $k \in A_p$ si $A_p \neq \emptyset$.

Le schéma suivant représente l'ordonnancement défini ci-dessus.

1er traitement



Lemme 2.2.1 -

Le système (2.2.1) respecte toutes les contraintes définies précédemment, en particulier : engagement journalier de chaque chaîne, tout objet passe dans le 1er traitement au plus tard la veille de son passage dans le 2ème traitement ; de plus le système (2.2.1) respecte le principe (1.1) : travail par part.

Démonstration -

Soit $k \in \{1, \dots, p-d_0\}$ kième jour de 2ème traitement.

L'engagement journalier de 2ème traitement s'écrit (regarder I.III.A), le système 1.3.2 :

$$Q_k = \sum_{l=1}^k Q_l = \sum_{l=1}^k \sum_{i=1}^n x_{li} \cdot b_i =$$

$$\sum_{l=1}^k \sum_{\substack{j=1 \\ j \neq i}}^n x_{lj} \cdot b_j + \sum_{l=1}^k \sum_{i=1}^n x_{li} \cdot b_i \quad \text{ou encore}$$

$$\sum_{l=1}^k \sum_{A^i} x_{lj} \cdot b_j + \sum_{l=1}^k \sum_{i=1}^n x_{li} \cdot b_i ;$$

Rappelons que $|p_j| \cdot b_j = \sum_{l=1}^p x_{lj} \cdot b_j \geq \sum_{l=1}^k x_{lj} \cdot b_j$

d'où : $\sum_{l=1}^k \sum_{A^i} x_{lj} \cdot b_j \leq \sum_{A^i} |p_j| \cdot b_j$

or le système (2.2.1) impose que :

$$\sum_{A^i} |p_j| \cdot b_j + \sum_{l=1}^k \sum_{i=1}^n x_{li} \cdot b_i \leq (d_0 + k-1) \times S \quad \text{donc}$$

$$\sum_{l=1}^k Q_l \leq (d_0 + k-1) \times S \quad \text{par conséquent l'engagement journalier}$$

de 2ème traitement est respecté. De même l'engagement journalier moyen de 1er traitement est respecté puisqu'on travaille S unités de temps.

Le système (2.2.1) respecte le principe (1.1), travail par part, en effet pour $k=1$ (2.2.1) s'écrit :

$$\sum_{A_i} |p_j| \cdot b_j + x_{1i} \cdot b_i \leq d_0 \times S$$

On constate que dans les d_0 premiers jours de 1er traitement, on exige de traiter toutes les parts entièrement et x_{1i} de p_i ième part, donc le temps de transition dans les d_0 premiers jours est égal à : $(\text{card } A - 1) \times C$, ensuite, dès qu'on a fini de travailler les objets de p_i on enchaîne avec les dernières parts de A_p . Nous avons un temps de transition de passage d'un objet de la part p_i à un objet de la part p_k , $k \in A_p$ si $A_p \neq \emptyset$. De plus, le temps de transition pour les parts p_k , $k \in A_p$ est $(\text{Card } A_p - 1) \times C$, au total :

$$\begin{aligned} & (\text{card } A - 1) \times C + 1 \times C + (\text{Card } A_p - 1) \times C = \\ & (\text{card } A + \text{card } A_p + 1 - 2) \times C = (\text{Card } (A \cup A_p) - 1) \times C \\ & = (\text{Card } A - 1) \times C \\ & = (n-1) \times C \end{aligned}$$

La démonstration est valable même $A_p = \emptyset$, en effet si $A_p = \emptyset$, le 1er traitement se termine avec les objets de la part p_i , et il n'y aura pas de temps de transition entre un objet de la part p_i et un objet d'une part p_k , $k \in A_p$, puisque $A_p = \emptyset$ et le temps de transition reste $(\text{card } A - 1) \times C = (n-1) \times C$.

N.B. - On vient de donner la justification du système (2.1.2) et la justification de 2/ du théorème (2.1).

Définition 2.2 - (Système d'ordre 1)

a) le système d'ordre 1 n'a pas de sens que :

- s'il existe un entier $j_0 \in \{2, \dots, p-1\}$ tel que

$j_0 = \min_{j=2}^{p-1} \{j, A_j \neq \emptyset\}$. d_1 est l'entier minimum qui vérifie :

$$(2.2.2) \quad \left\{ \begin{array}{l} \exists i \in A_1 \\ \forall k \in \{1, \dots, j_0 - 1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{l=1}^k x_{li} \cdot b_i \leq (d_1 + k - 1) \cdot S \\ \exists i \in B_{j_0} = A - A_1 \\ \forall k \in \{j_0, \dots, p - d_1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{B_{j_0}} |p_j| \cdot b_j + \sum_{l=j_0}^k x_{li} \cdot b_i \leq (d_1 + k - 1) \cdot S \end{array} \right.$$

Lemme 2.2.2. -

C'est l'énoncé du lemme (2.2.1) pour le système (2.2.2).

Démonstration -

Rappelons les points essentiels de la démonstration du lemme (2.2.1) en l'adaptant au système (2.2.2).

Engagement journalier :

pour $k \in \{1, \dots, j_0 - 1\}$, il suffit de remplacer A par A_1 et A^i par A_1^i ,

pour $k \in \{j_0, \dots, p - d_1\}$, il suffit de remarquer que

$$\forall j \in B_{j_0} = A - A_1, \quad x_{1j} = x_{2j} = \dots = x_{j_0-1,j} = 0 \quad \text{et}$$

Le premier élément non nul de p_j (peut apparaître au plus tôt) est $x_{j_0,j}$

ce qui justifie l'indice inférieur de la somme $\sum_{l=j_0}^k x_{li}$ pour $k \in \{j_0, \dots, p - d_1\}$. La démonstration reste encore valable en remplaçant A par $A_1 \cup B_{j_0}$ et A^i par $(A_1 \cup B_{j_0}^i)$.

Le principe (1.1) est respecté :

Remarquons que dans les d_1 premiers jours nous avons le nombre de changement en part (temps de transition) est $(\text{card } A_1 - 1)$.

Dans les $(d_1 + j_0 - 1)$ premiers jours nous avons en tout $(\text{card } A_1 - 1) \times C + 1 \times C + (\text{card } B_{j_0} - 1) \times C = (\text{card } A - 1) \times C$
Même chose si $A_p \neq \emptyset$.

Définition (2.2.3) - (Système d'ordre s , $1 \leq s \leq p-1$)

a) Le système d'ordre $s \geq 1$ n'a pas de sens que s'il existe des entiers positifs L_h définis ainsi :

$$L_1 = \min_{j=2}^{p-1} \{j, A_j \neq \emptyset\}$$

et pour h , $2 \leq h \leq s \leq p-1$

$$L_h = \min_{j=L_{h-1}+1}^{p-1} \{j, A_j \neq \emptyset\}$$

d_s est l'entier minimum qui vérifie :

$$\begin{aligned}
 & \exists i \in A_1 \\
 & \forall k \in \{1, \dots, L_1-1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{l=1}^k x_{li} \cdot b_i \leq (d_s + k - 1) \cdot S \\
 & \exists i \in A_{L_1} \\
 & \forall k \in \{L_1, \dots, L_2-1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{A_{L_1}} |p_j| \cdot b_j + \sum_{l=L_1}^k x_{li} \cdot b_i \leq (d_s + k - 1) \cdot S \\
 & \vdots \\
 & \exists i \in A_{L_h}, \quad h < s-1 \\
 & \forall k \in \{L_h, \dots, L_{h+1}-1\} \quad \left(\sum_{A_1 \cup \dots \cup A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{A_{L_h}} |p_j| \cdot b_j + \sum_{l=L_h}^k x_{li} \cdot b_i \right) \\
 & \leq (d_s + k - 1) \cdot S \\
 & \vdots \\
 & \exists i \in A_{L_{s-1}} \\
 & \forall k \in \{L_{s-1}, \dots, L_s-1\} \quad \left(\sum_{A_1 \cup \dots \cup A_{L_{s-2}}} |p_j| \cdot b_j + \sum_{A_{L_{s-1}}} |p_j| \cdot b_j + \sum_{l=L_{s-1}}^k x_{li} \cdot b_i \right) \\
 & \leq (d_s + k - 1) \cdot S \\
 & \exists i \in B_{L_s} = A - (A_1 \cup A_2 \cup \dots \cup A_{L_h} \cup \dots \cup A_{L_{s-1}}) \\
 & \forall k \in \{L_s, \dots, p-d_s\} \quad \left(\sum_{A_1 \cup \dots \cup A_{L_{s-1}}} |p_j| \cdot b_j + \sum_{B_{L_s}} |p_j| \cdot b_j + \sum_{l=L_s}^k x_{li} \cdot b_i \right) \\
 & \leq (d_s + k - 1) \cdot S
 \end{aligned}$$

N.B. - Si L_{s+1} n'existe pas, on peut remplacer B_{L_s} par A_{L_s} puisque dans ce cas $A = A_1 \cup A_2 \cup \dots \cup A_{L_s} \cup A_p$.

b) L'ordonnancement de 1er traitement associé au système (2.2.3)

est :

- $p_j, j \in A_1^i$ puis $p_i, i \in A_1 - A_1^i$, ensuite
- $p_j, j \in A_{L_1}^i$ puis $p_i, i \in A_{L_1} - A_{L_1}^i$ ensuite
- $\vdots$
- $h < s-1$ - $p_j, j \in A_{L_h}^i$, puis $p_i, i \in A_{L_h} - A_{L_h}^i$, ensuite
- $\vdots$
- $p_j, j \in B_{L_s}^i$, puis $p_i, i \in B_{L_s} - B_{L_s}^i$, et on termine par
- $p_j, j \in A_p$ si $A_p \neq \emptyset$.

Théorème 2.2. -

Le système d'ordre s (2.2.3) vérifie le principe (1.1) travail par part,

- (T_2) l'ordonnancement de 2ème traitement qui est déjà fixé,
- (M_3) toutes les chaînes en état de travail,
- (M_4) l'engagement journalier de chaque chaîne est respecté,
- (OM_1 et OM_2) un objet qui passe par le 2ème traitement doit être passé par le 1er traitement l'un des jours précédent (et pas le jour même).

Démonstration -

Le système (2.2.3) respecte le principe de travail par part, en effet il suffit de regarder l'ordonnancement associé à ce système, on peut indiquer le nombre de changements en part à chaque date de 1er traitement :

- à la fin de d_s ième jours de 1er traitement, nous avons ($\text{card } A_1 - 1$) changements,
- à la fin de $(d_s + L_1 - 1)$ ième jours de 1er traitement, nous avons ($\text{card } (A_1 \cup A_{L_1}) - 1$) changements,
- $\vdots$

- à la fin de $(d_s + L_h - 1)$ ième jours de 1er traitement, nous avons $(\text{card}(A_1 \cup A_{L_1} \cup \dots \cup A_{L_h}) - 1)$ changements,

⋮

- à la fin de $(d_s + L_s - 1)$ ième jours de 1er traitement, nous avons $\text{card}(A_1 \cup A_{L_1} \cup \dots \cup A_{L_{s-1}} \cup B_{L_s} - 1)$ changements et dès qu'on termine le traitement de la part p_i vérifiant la dernière inégalité de système, on traite les parts p_k , $k \in A_p$ si $A_p \neq \emptyset$. Puisque les ensembles $A_1, A_{L_1}, \dots, A_{L_h}, \dots, B_{L_s}, A_p$ forment une partition de A , nous avons $(\text{card}(A_1 \cup \dots \cup B_{L_s} \cup A_p) - 1) = (n - 1)$ changements. Donc le temps de transition minimum est atteint à savoir $(n - 1) \times C$ et le critère (C1.1) est satisfait.

De plus, le système (2.2.3) est identique au système (1.3.2) qui vérifie toutes les contraintes déjà citées. Rappelons brièvement que l'engagement journalier de 2ème traitement est respecté pour un $k^{\text{ième}}$ jour : $k \in \{L_h, \dots, L_{h+1} - 1\}$. Montrons que $\sum_{l=1}^k (x_{l1} \cdot b_1 + \dots + x_{ln} \cdot b_n) \leq (d_s + k - 1) \times S$.

Remarquons d'abord que pour $i \notin (A_1 \cup A_{L_1} \cup \dots \cup A_{L_{h-1}} \cup A_{L_h})$ nous avons : $x_{li} = 0$ pour $l \in \{1, \dots, L_1, L_1 + 1, \dots, L_h, \dots, L_{h+1} - 1\}$.

Alors pour $k \leq L_{h+1} - 1$ nous avons :

$$\begin{aligned} \sum_{l=1}^k (x_{l1} \cdot b_1 + \dots + x_{ln} \cdot b_n) &= \sum_{A_1 \cup \dots \cup A_{L_{h-1}}} \sum_{l=1}^k x_{lj} \cdot b_j + \sum_{A_{L_h}^i} \sum_{l=L_n}^k x_{lj} \cdot b_j + \sum_{l=L_n}^k x_{li} \cdot b_i \\ &\leq \sum_{A_1 \cup \dots \cup A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{A_{L_h}^i} |p_j| \cdot b_j + \sum_{l=L_h}^k x_{li} \cdot b_i \leq (d_s + k - 1) \cdot S \end{aligned}$$

La démonstration est valable pour tout $k \in \{1, \dots, p - d_s\}$.

c/ Calcul numérique de d_s , $0 \leq s \leq p - 1$ -

La méthode utilisée est identique à celle de la solution du système (1.3.3).

Etape 1 -

pour $k \in \{1, \dots, L_1 - 1\}$, soit i_0 fixé, $i_0 \in A$,

soit $k_o \in \{1, \dots, L_1-1\}$ tel que :

$$d_{k_o}^{i_o} = \max_{k=1}^{L_1-1} \{ \lceil (\sum_{j=1}^n |p_j| \cdot b_j + \sum_{l=1}^k x_{l i_o} \cdot b_{i_o}) : \bar{S} \rceil + 1 - k \}$$

soit $i_{A_1} \in A_1$ tel que $d_{k_o}^{i_{A_1}} = \min_{i_o \in A_1} \{ d_{k_o}^{i_o} \}$

Etape 2 -

D'une manière générale pour $k \in \{L_h, \dots, L_{h+1}-1\}$, $h < s-1$

Soit $i_o \in A_{L_h}$, $k_o \in \{L_h, \dots, L_{h+1}-1\}$ tel que :

$$d_{k_o}^{i_o} = \max_{k=L_h}^{L_{h+1}-1} \{ \lceil (\sum_{j=1}^n |p_j| \cdot b_j + \sum_{i_o \in A_{L_h}} |p_j| \cdot b_j + \sum_{l=L_h}^k x_{l i_o} \cdot b_{i_o}) : \bar{S} \rceil + 1 - k \}$$

soit $i_{A_{L_h}} \in A_{L_h}$ tel que $d_{k_o}^{i_{A_{L_h}}} = \min_{i_o \in A_{L_h}} \{ d_{k_o}^{i_o} \}$

Etape 3 -

pour $k \in \{L_s, \dots, p-1\}$, soit $i_o \in B_{L_s}$ et $k_o \in \{L_s, \dots, p-1\}$

tel que :

$$d_{k_o}^{i_o} = \max_{k=L_s}^{p-1} \{ \lceil (\sum_{j=1}^n |p_j| \cdot b_j + \sum_{i_o \in B_{L_s}} |p_j| \cdot b_j + \sum_{l=L_s}^k x_{l i_o} \cdot b_{i_o}) : \bar{S} \rceil - k + 1 \}$$

soit $i_{B_{L_s}} \in B_{L_s}$ tel que :

$$d_{k_o}^{i_{B_{L_s}}} = \min_{i_o \in B_{L_s}} \{ d_{k_o}^{i_o} \}$$

Etape 4 -

$$d_s = \text{Max} \{ d_{k_o}^{i_{A_1}}, d_{k_o}^{i_{A_{L_1}}}, \dots, d_{k_o}^{i_{A_{L_h}}}, \dots, d_{k_o}^{i_{A_{L_{s-1}}}}, d_{k_o}^{i_{A_{L_s}}} \}$$

d'après le théorème (1.1) d_s est l'entier minimum vérifiant le système d'ordre s .

Conséquence 2.1. -

Soit d_h l'entier minimum vérifiant le système d'ordre h , $1 \leq h \leq s-1$, avec l'ordre s le dernier système qui peut exister, c'est-à-dire :

$$A = A_1 \cup A_{L_1} \cup A_{L_2} \cup \dots \cup A_{L_h} \cup A_{L_{h+1}} \cup A_{L_s} \cup A_p$$

$$\text{Rappelons que } d_h = \text{Max} \{ d_{k_o}^{i_{A_1}}, \dots, d_{k_o}^{i_{A_{L_{h-1}}}}, d_{k_o}^{i_{B_{L_h}}} \}$$

Supposons que $d_h \leq d_{h-1} \leq \dots \leq d_1$

$$\text{Alors si } \underline{d_{k_o}^{i_{A_{L_h}}} > d_h} \quad \text{alors pour tout } k \in \{h+1, \dots, s\}$$

$$\text{nous avons } \underline{d_k > d_h}$$

Démonstration -

Il suffit de remarquer que :

$$d_{h+1} = \text{Max} \{ d_{k_o}^{i_{A_1}}, \dots, d_{k_o}^{i_{A_{L_{h-1}}}}, d_{k_o}^{i_{A_{L_h}}}, d_{k_o}^{i_{B_{L_{h+1}}}} \}$$

$$\text{donc : } d_{h+1} > d_h \quad \text{car } d_{h+1} \geq d_{k_o}^{i_{A_{L_h}}}$$

Il en est de même pour k , $h+1 \leq k \leq s$,

$$\text{car } d_k \geq d_{k_o}^{i_{A_{L_h}}} > d_h$$

Remarque -

Il en est de même pour d_0 et d_1 ,

$$\text{si } d_{k_0}^{iA_1} > d_0, \text{ alors } d_1 = \text{Max} \{ d_{k_0}^{iA_1}, d_{k_0}^{BL_1^i} \} > d_0$$

et il est inutile de calculer d_k entier minimum vérifiant le système d'ordre k , $1 \leq k \leq s$.

III - SOLUTION DU PROBLEME $n|m|1,1|D1,(D2,T2)|(C1.1)$ et C2 -

Dans ce paragraphe nous allons décrire un algorithme qui donnera le nombre de jours de décalage le plus petit entre les deux traitements, dont chacun a une seule chaîne, en effet, tous les systèmes décrits précédemment donnent le temps de transition minimum pour le 1er traitement.

Pour simplifier l'algorithme, nous allons ignorer la conséquence (2.1) pour l'instant, car on donnera plus tard la programmation structurée qui tiendra compte de tous les tests d'arrêt de calcul.

Soit $d_0, d_1, \dots, d_s$ les entiers minimum vérifiant les systèmes d'ordre $0, 1, \dots, s$ pour l'ensemble $A = A_1 \cup A_{L_1} \cup \dots \cup A_{L_s} \cup A_p$, $\text{card } A = n > 1$, avec le système d'ordre 0, on obtient un d_0 jour de décalage, dans lequel on exige de faire $(\text{card}(A_1 \cup A_{L_1} \cup \dots \cup A_{L_s}) - 1)$ changements en parts. Pour nous, les d_0 jours, c'est le maximum^s qu'on puisse avoir comme nombre de jours de décalage, le but quand on passe d'un système d'ordre h à un système d'ordre $(h+1)$. On espère diminuer le nombre de jours de décalage, si on trouve $d_{h+1} > d_h$ avec une condition bien déterminée, il est inutile de calculer les $d_{h+2}, \dots, d_s$, d'où l'idée de l'algorithme suivant.

Algorithme 2.3.1. -

On note par I l'ordre du système à déterminer, alors d , qui est le nombre de jours de décalage le plus petit entre les deux traitements, est déterminé ainsi :

DEBUT

1. $d = d_0$; $I = 0$; $h = 1$;
2. TANTQUE $h \leq s$ FAIRE

DEBUT

SI $d_h > d$ ALORS DEBUTSI $h \leq s-1$ ALORS DEBUT
 SI $\sum_{j=h}^s |p_j| \cdot b_j < (L_{h+1} - L_h + 1) \cdot S$ ALORS

DEBUT

 $d = d_{h-1}$; $I = h-1$; $h = s+1$;

FIN

SINON $h = h+1$;FINSINON $h = h+1$;FINSINON DEBUT $d = d_h$; $I = h$; $h = h+1$;FIN ;

FIN ;

FIN ;

Démonstration -

L'algorithme donne $d = \min_{h=0}^s d_h$ sauf dans le cas où

$d_h > d_{h-1}$ et $\sum_{j=h}^s |p_j| \cdot b_j < (L_{h+1} - L_h + 1) \cdot S$, dans ce cas

montrons que d_{h+1} , d_{h+2} jusqu'à d_s sont supérieurs à d_{h-1} . De plus, le résultat est évident si $n \leq 2$ ceci pour tout p fixé dans $\mathbb{N}^*$, pour que d_{h-1} , d_h et d_{h+1} existe il faut que $n \geq 3$, si $n \geq 3$ il faut que $p \geq 4$ sinon le résultat est évident.

Donc on suppose que $n \geq 3$ et $p \geq 4$ (p = nombre de jours de travail de la période). Pour mieux comprendre la démonstration, nous allons la faire

en deux étapes, d'abord pour $n = 3$, puis pour $n > 3$; la démonstration se généralise facilement.

Etape 1 -

Nous avons $n = 3$, donc $A = \{1, 2, 3\}$ associé à p_1, p_2, p_3 .
On considère que $A_p = \emptyset$ sinon on aura pas trois systèmes. Nous sommes dans le cas où $d_1 > d_0$, montrons que d_2 ne peut pas être strictement supérieur à d_0 si :

$$\sum_{A_{L_1} \cup A_{L_2}} |p_j| \cdot b_j < (L_2 - L_1 + 1) \times S.$$

Rappelons d'abord les trois systèmes pour $A = \{1, 2, 3\}$.

Système d'ordre 0 -

Soit d_0 l'entier minimum vérifiant :

$$\exists i \in A$$

$$\forall k \in \{1, \dots, p - d_0\} \quad \sum_{A_0} |p_j| \cdot b_j + \sum_{l=1}^k x_{li_0} \cdot b_{i_0} \leq (d_0 + k - 1) \cdot S$$

Système d'ordre 1 -

Soit $A_1 = \{1\}$, on sait qu'il existe $L_1 \in \{2, \dots, p-1\}$; $A_{L_1} \neq \emptyset$

avec $L_1 = \min_{j=2}^{p-1} \{j, A_j \neq \emptyset\}$,

Soit d_1 l'entier minimum vérifiant :

$$(2.1) \quad \begin{aligned} &\exists i \in A_1 \quad (i=1) \\ &\forall k \in \{1, \dots, L_1 - 1\} \quad \sum_{l=1}^k x_{li_1} \cdot b_{i_1} \leq (d_1 + k - 1) \cdot S \end{aligned}$$

$$\exists j_0 \in B_{L_1} = \{2, 3\}$$

$$(2.2) \quad \forall k \in \{L_1, \dots, p - d_1\} \quad |p_1| \cdot b_1 + \sum_{\substack{k \\ k \neq j_0}} |p_k| \cdot b_k + \sum_{l=1}^k x_{lj_0} \cdot b_{j_0} \leq (d_1 + k - 1) \cdot S$$

Système d'ordre 2 -

Soit $A_1 = \{1\}$, $A_{L_1} = \{2\}$, $A_{L_2} = \{3\}$ avec $L_2 = \min_{j=L_1+1}^{p-1} \{j, A_j \neq \emptyset\}$

Soit d_2 l'entier minimum vérifiant :

$$\begin{aligned} \exists i \in A_1 \quad (i=1) \\ \forall k \in \{1, \dots, L_1-1\} \quad \sum_{l=1}^k x_{li} \cdot b_l \leq (d_2+k-1) \times S \end{aligned} \quad \dots (3.1)$$

$$\begin{aligned} \exists i \in A_{L_1} \quad (i=2) \\ \forall k \in \{L_1, \dots, L_2-1\} \quad |p_1| \cdot b_1 + \sum_{l=L_1}^k x_{li} \cdot b_l \leq (d_2+k-1) \times S \end{aligned} \quad \dots (3.2)$$

$$\begin{aligned} \exists i \in A_{L_2} \quad (i=3) \\ \forall k \in \{L_2, \dots, p-d_2\} \quad |p_1| \cdot b_1 + |p_2| \cdot b_2 + \sum_{l=L_2}^k x_{li} \cdot b_l \leq (d_2+k-1) \times S \end{aligned} \quad \dots (3.3)$$

Montrons d'abord que $i_0 \in \{2, 3\}$ c'est-à-dire $i_0 = 1$

a) On suppose que (3.1) est vrai sinon la démonstration est terminée ;

b) si j_0 dans (2.2) est tel que $i_0 = j_0$, cela implique que l'hypothèse $d_1 > d_0$ est fautive, en effet pour $i \in \{2, 3\}$ nous avons pour tout $l \in \{1, \dots, L_1-1\}$ $x_{li} = 0$ et (2.2) s'écrit pour $i_0 = j_0$:

$$\begin{aligned} \forall k \in \{L_1, \dots, p-d_1\} \quad |p_1| \cdot b_1 + |p_k| \cdot b_k + \sum_{l=1}^k x_{lj_0} \cdot b_{j_0} \leq (d_0+k-1) \cdot S \\ \leq (d_1+k-1) \cdot S \end{aligned}$$

et ceci contredit l'hypothèse que d_1 est l'entier minimum vérifiant le système, donc dans (1.1) $i_0 = 1$.

D'où, d'après (1.1) et la condition imposée par l'algorithme nous avons

$$(**) \quad |p_2| \cdot b_2 + |p_3| \cdot b_3 < (L_2 - L_1 + 1) \cdot S \quad (\text{ici } h=1)$$

d'après (3.2) et pour $k=L_1$ nous avons :

$$(***) \quad |p_1| \cdot b_1 + x_{L_1 2} \cdot b_2 \leq (d_2 + L_1 - 1) \cdot S,$$

or le premier traitement se fait aussi en p jours, et le $p^{\text{ième}}$ jours de 1er traitement correspond à la date $(p-d_2)^{\text{ième}}$ jours de 2ème traitement.

Donc d'après (***) pour faire $\sum_{l=L_1+1}^p x_{l2} \cdot b_2 + |p_3| \cdot b_3$ on travaille

les jours suivants,

(d_2+L_1) ième jours de 1er traitement	→	L_1 ième jour de 2ème traitement
(d_2+L_1+1) " " "	→	(L_1+1) " " "
⋮		
(d_2+L_1) " " "	→	L_2 " " "
(d_2+L_2+1) " " "	→	(L_2+1) " " "
⋮		
p ième jours de 1er traitement	→	$(p-d_2)$ ième jours de 2ème traitement

Rappelons que $x_{L_1 2}$ est faite à la date (d_2+L_1-1) , ces nombres de jours pour faire $|p_2| \cdot b_2 + |p_3| \cdot b_3$ est supérieure ou égal à (L_2-L_1+1) et ceci contraire à (**), donc $d_2 > d_0$.

Etape 2 - (généralisation pour $n > 3$).

(α).... $d_h > d_{h-1}$ et $\sum_{A_L \cup \dots \cup A_{L_s} \cup A_p} |p_j| \cdot b_j < (L_{h+1}-L_h+1) \cdot S$

Nous allons montrer que $d_{h+1}, \dots, d_s$ sont supérieurs à d_{h-1} , pour $d_{h+2}, \dots, d_s$. C'est évident, il suffit de remarquer que :

(β).... $L_s-L_h+1 > L_{s-1}-L_h+1 > \dots > L_{h+2}-L_{h+1}+1 > L_{h+1}-L_h+1$

Par conséquent la démonstration est finie si on montre que $d_{h+2} > d_{h-1}$.

La démonstration pour $n=3$ s'applique ici, rappelons la avec l'écriture du système d'ordre $h-1, h, h+1$.

Le système d'ordre h-1 -

Soit d_{h-1} l'entier minimum vérifiant :

$$\exists i \in A_1$$

$$(0, h-1) \dots \forall k \in \{1, \dots, L_1-1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{l=1}^k x_{li} \cdot b_i \leq (d_{h-1} + k - 1) \cdot S$$

$$\exists i \in A_{L_1}$$

$$(1, h-1) \dots \forall k \in \{L_1, \dots, L_2-1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{A_{L_1}} |p_j| \cdot b_j + \sum_{l=L_1}^k x_{li} \cdot b_i \leq (d_{h-1} + k - 1) \cdot S$$

$$\vdots$$

$$\vdots$$

$$\exists i_o \in B_{L_{h-1}} = A - (A_1 \cup \dots \cup A_{L_{h-2}})$$

$$(h-1, h-1) \quad \forall k \in \{L_{h-1}, \dots, p-d_{h-1}\} \quad \sum_{A_1 \cup \dots \cup A_{L_{h-2}}} |p_j| \cdot b_j + \sum_{B_{L_{h-1}}} |p_j| \cdot b_j + \sum_{l=L_{h-1}}^k x_{li_o} \cdot b_{i_o} \leq (d_{h-1} + k - 1) \cdot S$$

Le système d'ordre h -

Soit d_h l'entier minimum vérifiant :

$$(0, h) = (0, h-1) \dots \leq (d_h + k - 1) \cdot S$$

$$\vdots$$

$$(h-2, h) = (h-2, h-1) \dots \leq (d_h + k - 1) \cdot S$$

$$\exists i \in A_{L_{h-1}}$$

$$(h-1, h) \dots \forall k \in \{L_{h-1}, \dots, L_h-1\} \quad \sum_{A_1 \cup \dots \cup A_{L_{h-2}}} |p_j| \cdot b_j + \sum_{A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{l=L_{h-1}}^k x_{li} \cdot b_i \leq (d_h + k - 1) \cdot S$$

$$\exists j_o \in B_{L_h}$$

$$(h, h) \dots \forall k \in \{L_h, \dots, p-d_h\} \quad \sum_{A_1 \cup \dots \cup A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{B_{L_h}} |p_j| \cdot b_j + \sum_{l=L_h}^k x_{lj_o} \cdot b_{j_o} \leq (d_h + k - 1) \cdot S$$

Le système d'ordre h+1 -

Soit d_{h+1} le plus entier vérifiant :

$$(0, h+1) = (0, h) = (0, h-1) \quad \dots \leq (d_{h+1} + k - 1) \cdot S$$

$$(1, h+1) = (1, h) = (1, h-1) \quad \dots \leq (d_{h+1} + k - 1) \cdot S$$

$\vdots$

$$(h-2, h+1) = (h-2, h) = (h-2, h-1) \quad \dots \leq (d_{h+1} + k - 1) \cdot S$$

$$(h-1, h+1) = (h-1, h) \quad \dots \leq (d_{h+1} + k - 1) \cdot S$$

$$\exists i \in A_{L_h}$$

$$(h, h+1) \dots \forall k \in \{L_h, \dots, L_{h+1}-1\} \quad \sum_{A_1 U \dots U A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{A_{L_h}^i} |p_j| \cdot b_j + \sum_{l=L_h}^k x_{li} \cdot b_i \leq (d_{h+1} + k - 1) \cdot S$$

$$\exists k_0 \in B_{L_{h+1}}$$

$$(h+1, h+1) \dots \forall k \in \{L_{h+1}, \dots, p-d_{h+1}\} \quad \sum_{A_1 U \dots U A_{L_h}} |p_j| \cdot b_j + \sum_{B_{L_{h+1}}^{k_0}} |p_j| \cdot b_j + \sum_{l=L_{h+1}}^k x_{lk_0} \cdot b_{k_0} \leq (d_{h+1} + k - 1) \cdot S$$

Comme dans l'étape 1 :

a) on suppose que les inégalités $(0, h+1)$; $(1, h+1), \dots, (h-1, h+1)$ sont vraies, sinon la démonstration est terminée, c'est-à-dire $d_{h+1} < d_{h-1}$,

b) Montrons maintenant que $\forall j_0 \in B_{L_h}, j_0 \neq i_0$ pour $i_0 \in B_{L_{h-1}}$.

Rappelons d'abord que pour $i \in B_{L_h}$, et pour $k=1, \dots, L_h-1$ nous avons $x_{li}=0$, en particulier pour $k=L_{h-1}, \dots, L_h-1$; donc si j_0 vérifie (h, h) est égal à i_0 , on pourrait écrire pour tout

$$k \in \{L_h, \dots, p-d_h\} \quad \sum_{A_1 U \dots U A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{B_{L_h}^{j_0}} |p_j| \cdot b_j + \sum_{l=L_h}^k x_{lj_0} \cdot b_{j_0} \leq (d_{h-1} + k - 1) \cdot S$$

Comme nous avons supposé $d_{h+1} < d_{h-1} < d_h$ de plus $(0,h), \dots, (h-1,h)$ sont vérifiés pour d_{h-1} , donc on vient de mettre en évidence un $d=d_{h-1}$ inférieur strictement à d_h et vérifie le système d'ordre h , absurde. Donc $\forall j_0 \in B_{L_h}$ $j_0 \neq i_0$. Maintenant d'après $(h,h+1)$ et en particulier pour $k=L_h$, nous avons :

$$\sum_{A_1 \cup \dots \cup A_{L_{h-1}}} |p_j| \cdot b_j + \sum_{A_{L_h}^i} |p_j| \cdot b_j + x_{L_h i} \cdot b_i \leq (d_{h+1} + L_h - 1) \cdot S$$

$$1/ \sum_{l=L_h+1}^p x_{li} + \sum_{A_{L_{h+1}}} \dots \sum_{A_{L_s}} \sum_{A_p} |p_j| \cdot b_j \leq \sum_{A_{L_h}} |p_j| \cdot b_j + \sum_{A_{L_{h+1}}} \dots \sum_{A_{L_s}} \sum_{A_p} |p_j| \cdot b_j$$

$$2/ \sum_{l=L_h+1}^p x_{li} \cdot b_i + \sum_{A_{L_{h+1}}} \dots \sum_{A_p} |p_j| \cdot b_j \text{ est fait pendant les jours suivants.}$$

$$\begin{array}{llll} (d_{h+1} + L_h) \text{ième jours de 1er traitement} & \rightarrow & L_h \text{ième jours de 2ème traitement} \\ \vdots & & \\ (d_{h+1} + L_{h+1}) & " & " & " \rightarrow L_{h+1} & " & " & " \\ \vdots & & & \\ p \text{ième jours de 1er traitement} & \rightarrow & p - d_{h+1} & " & " & " \end{array}$$

et ce nombre de jours est strictement supérieur à $(L_{h+1} - L_h + 1)$. Alors (α) est fausse, ce qui est absurde. Donc $d_{h+1} > d_{h-1}$, avec (β) la démonstration est finie.

Conséquence 2.3.1. -

Avec les mêmes notations de paragraphe II et III de ce chapitre, alors si

$$\sum_{A_1} |p_j| \cdot b_j \leq \sum_{A_{L_1}} \dots \sum_{A_{L_s}} \sum_{A_p} |p_j| \cdot b_j$$

nous avons $d_1 \leq d_0$.

Démonstration -Rappel -

Soit d_0 l'entier minimum vérifiant :

$$\exists i \in A$$

$$\forall k \in \{1, \dots, p-d_0\} \quad \sum_{A^i} |p_j| \cdot b_j + \sum_{l=1}^k x_{li} \cdot b_i \leq (d_0+k-1) \cdot S \quad \dots \quad (1.1)$$

Soit d_1 l'entier minimum vérifiant :

$$\exists i_1 \in A_1$$

$$\forall k \in \{1, \dots, L_1-1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{l=1}^k x_{li_1} \cdot b_{i_1} \leq (d_1+k-1) \cdot S \quad \dots \quad (2.1)$$

$$\exists i_2 \in B_{L_1} = A_{L_1} \cup \dots \cup A_{L_s}$$

$$\forall k \in \{L_1, \dots, p-d_1\} \quad \sum_{A_1} |p_j| \cdot b_j + \sum_{B_{L_1}} |p_j| \cdot b_j + \sum_{l=1}^k x_{li_2} \cdot b_{i_2} \leq (d_1+k-1) \cdot S \quad \dots \quad (2.2)$$

Maintenant supposons que $d_1 > d_0$ avec $\sum_{A_1} \leq \sum_{A_{L_1} \cup \dots \cup A_{L_s} \cup A_p}$

a) Soit $i \in B_{L_1}$ et i vérifiant (1.1).

Puisque dans (1.1) $i \in B_{L_1}$ alors $A_1 \subset A^i$ donc $\sum_{A_1} |p_j| \cdot b_j < d_0 \times S$
De plus, pour tout $i_1 \in A_1$ nous avons :

$$\sum_{A_1} |p_j| \cdot b_j + \sum_{l=1}^k x_{li_1} \cdot b_{i_1} \leq \sum_{A_1} |p_j| \cdot b_j < d_0 \times S$$

et ceci pour tout $k \in \{1, \dots, p\}$.

Donc (2.1) est vérifié par d_0 car $d_0 \times S \leq (d_0+k-1) \cdot S$ pour $k \in \{1, \dots, L_1-1\}$.

Rappelons que $i_2 \in B_{L_1}$ tel que $d^{i_2} = \min_{j \in B_{L_1}} \{d^j\}$ avec :

$$d^j = \max_{k=L_1}^{p-1} \left\{ \left\lceil \sum_{B_{L_1}^j} |p_i| \cdot b_i + \sum_{l=L_1}^k x_{lj} \cdot b_j \right\rceil + 1 - k \right\}$$

de même pour $i_1 \in A_1$: $d^{i_1}_1 = \min_{j \in A_1} \{d^j\}$

$d_1 = \max \{d^{i_1}_1, d^{i_2}_1\}$, on convient de montrer que $d^{i_1}_1 \leq d_0$ puisque d_0 vérifie (2.1).

i vérifiant (1.1) est tel que $d_0 = d^i = \min_{j \in A} \{d^j\} = \min_{j \in B_{L_1}} \{d^j\}$, car $i \in B_{L_1}$

or ici $d^j = \max_{k=1}^{p-1} \{ \left\lceil \sum_{A_j} + \sum_{l=1}^k x_{l_j} \cdot b_j \right\rceil : S \} + 1 - k$

Donc $d^{i_2}_1 \leq d_0$ par suite d_0 vérifie (2.2).

d_0 vérifie (2.1) et (2.2) avec d_1 le plus petit entier vérifiant le système d'ordre 1 ; cela contredit l'hypothèse $d_1 > d_0$, par conséquent

i vérifiant (1.1) n'appartient pas à B_{L_1} ($i \notin B_{L_1}$).

b) On suppose maintenant $i \in A_1$

d'après (1.1) $|p_i| \cdot b_i > (p - d_0) \times S$

d'après le système d'ordre 1, nous avons $\sum_{B_{L_1} \cup A_p} |p_j| \cdot b_j < (p - d_1) \times S$

nous avons considéré que $d_1 > d_0$ donc :

$p - d_0 > p - d_1$; d'où :

$\sum_{B_{L_1} \cup A_p} |p_j| \cdot b_j < (p - d_1) \cdot S < (p - d_0) \cdot S < |p_i| \cdot b_i \leq \sum_{A_1} |p_j| \cdot b_j$ absurde

car l'hypothèse est $\sum_{A_1} \leq \sum_{B_{L_1} \cup A_p}$ donc $i \notin A_1$. Dans a) $i \notin B_{L_1}$, dans

b) $i \notin A_1$ donc $i \notin A$ ce qui est absurde ; donc l'hypothèse $d_1 > d_0$ est fautive, par conséquent $d_1 \leq d_0$.

Programmation structurée -Algorithme 2.3.2. -

C'est l'algorithme (2.3.1) avec les conséquences (2.1) et (2.3.1).

Notations -

$$d_{A_{L_h}} = \min_{i \in A_{L_h}} \{d_{k_o}^i\} ; \quad d_{B_{L_h}} = \min_{i \in B_{L_h}} \{d_{k_o}^i\} ; \text{ on note par}$$

$$A_{L_o} = A_1 ; \quad X = \sum_{A_1} |p_j| \cdot b_j ; \quad Y = \sum_{(A \cup A_p) - A_1} |p_j| \cdot b_j$$

$$Z = \sum_{A_{L_h} \cup \dots \cup A_{L_s}} |p_j| \cdot b_j ;$$

L'entier minimum d déterminé par l'algorithme (2.3.1) est déterminé ainsi :

Etape 1 - Déterminer l'ensemble A_1 et les ensembles $A_{L_1}, A_{L_2}, \dots, A_{L_s}, A_p$ s'ils existent.

Etape 2 - si $X > Y$ alors calculer d_o
 | sinon $d_o = p+1$;
 | $d = d_o, I = 0, h = 1$;

Etape 3 - Tantque $h \leq s$ faire

DEBUT

 . calculer $d_{A_{L_{h-1}}}$

si $d_{A_{L_{h-1}}} > d$ ALORS DEBUT
 | $d = d_{A_{L_{h-1}}}$; $I = h-1$; $h = s+1$;
 | FIN

SINON DEBUT

 . calculer $d_{B_{L_h}}$; $d_h = \max(d_{A_{L_{h-1}}}, d_{B_{L_h}})$;

si $d_h > d$ ALORS DEBUT

 si $h \leq s-1$ ALORS DEBUT

 SI $Z < (L_{h+1} - L_h + 1) \cdot S$ alors

 DEBUT

 | $d = d_h, I = h-1, k = s+1$;

 FIN ;

$h = h+1$

 FIN ;

```

      |
      | h=h+1 ;
      |
      | FIN
SINON DEBUT
      |
      | d=dh, I=h, h=h+1 ;
      |
      | FIN ;
      |
FIN ;
FIN ;

```

IV - APPLICATION -

Rappelons que nous sommes dans le cas où chaque traitement est composé d'une seule chaîne, le temps de transition ne dépend pas des peintures, c'est une constante C , le but est d'obtenir les $(n-1)$ changements en peinture, avec le nombre de jours de décalage le plus petit. Dans les six exemples suivants on considère pour tout p_i , $i \in A = \{1, 2, \dots, n\}$ la durée d'exécution d'un objet de la part p_i , b_i est égal à 1 unité de temps.

- EXEMPLES NUMERIQUES -

Exemple 1 -

Soit $n=3$, $A = \{1, 2, 3\}$ associé à p_1, p_2, p_3 , on considère que la période $p=5$ jours, soit la matrice $X=(x_{ji})$ représente l'engagement journalier fixé de 2ème traitement.

	p_1	p_2	p_3
1	200	100	500
2	300	100	400
3	500	100	400
4	499	0	400
5	1	0	1000

$|p_1| = 1500$ objets, $|p_2| = 3000$ objets,
 $|p_3| = 2700$ objets

engagement journalier moyen de la chaîne
 de peinture $S = \frac{4500 \times 1}{5}$

$S = 900$ unités de temps.

Etape 1 - $A_1 = A$, $A_{L_1} = \dots = A_{L_S} = A_p = \emptyset$ dans $s=0$

Etape 2 - $\sum_{A_1} |p_j| \cdot 1 > \sum_{\emptyset} = 0$ donc on calcule $d_0 = \min_{i \in A} \{d^i\}$ et

$$d^i = \max_{k=1}^{p-1} \left\{ \left\lceil \sum_{A^i} (p_j) x_1 + \sum_{l=1}^k x_{li} \cdot 1 \right\rceil : 900 \right\rceil + 1 - k \right\} = \max_{k=1}^{p-1} \{d_k^i\}$$

on trouve $d^1 = d_1^1 = 4$ jours ; $d^2 = d_1^2 = 5$ jours ; $d^3 = d_1^3 = d_2^3 = 3$ jours

Donc d_0 entier minimum vérifiant le système d'ordre 0 est $d_0 = d^3 = 3$;
pour $i=3$ et l'ordonnancement est : $p_i, i \in A^3$ puis p_3 .

L'étape 3 ne sera pas exécutée puisque $s=0$ et dans l'étape 2 : $h=1$;
 $h \nless s$.

Exemple 2 -

$n=4$ parts, $p=5$ jours, soit

la matrice $X = (x_{ji})$ représentant
l'engagement journalier fixé de 2ème
traitement avec :

$|p_1| = 20$ objets, $|p_2| = 4$ objets,

$|p_3| = 9$ objets, $|p_4| = 7$ objets.

$$s = \frac{40 \times 1}{5} = 8 \text{ unités}$$

	p_1	p_2	p_3	p_4
1	8	0	0	0
2	8	0	0	0
3	4	1	1	1
4	0	1	3	5
5	0	2	5	1

Etape 1 - $A_1 = \{1\}$

$A_{L_1} = \{2,3,4\}$ avec $L_1 = 3$, $A_p = \emptyset$ nous avons $s = 1$.

Etape 2 - $\sum_{A_1} |p_j| \cdot b_j = |p_1| \times 1 \leq (|p_2| + |p_3| + |p_4|) \times 1$, $20 \leq 20$ unité
temps, donc $d_0 = p + 1 = 6$, ainsi d_0 ne pourra jamais être la solution.

Etape 3 - $h = 1$, $d = d_0$

Comme $1 \leq 1$ on peut exécuter l'étape 3 une fois

$$\text{Calculons } d_{A_1} : d_{A_1} = \max_{k=1}^{L_1-1} \left\{ \left\lceil \sum_{l=1}^k x_{li} \right\rceil : S \right\rceil + 1 - k \right\} = \min_{i \in A_1 = \{1\}} \{d^i\}$$

d'où $d_{A_1} = 1$; en effet $x_{11} = 8$ et $x_{21} = 8$;

• $d_{A_1} \nmid d = d_0 = 6$ donc on peut calculer $d_{B_{L_1}}$

$$d_{B_{L_1}} = \min_{i \in \{2,3,4\}} \{d^i\}.$$

Pour $k=3$ et $i=2$ nous avons $|p_1| + |p_3| + |p_4| + x_{32} = 37$ d'où $d_3^2 = 3 = d^2$.

Pour $k=3$ et $i=3$ nous avons $|p_1| + |p_2| + |p_4| + x_{33} = 32$ d'où $d_3^3 = 2 = d^3$.

Pour $k=3$ et $i=4$ nous avons $|p_1| + |p_2| + |p_3| + x_{34} = 34$ d'où $d_3^4 = 3 = d^4$.

$$\text{Donc } d_{B_{L_1}} = d^3 = 2, d_1 = \max \{d_{A_1}, d_{B_{L_1}}\} = \{1, 2\} = 2$$

donc le nombre de jours de décalage est $d = d_1 = 2$; associé à l'ordonnement du système 1 :

d'abord la peinture p_1 ; puis p_i , $i \in B_{L_1}^3$ puis p_3 .

Exemple 3 -

$n=4$ parts, $p=5$ jours, soit la matrice $X = (x_{ji})$ représentant l'engagement journalier fixé de 2ème traitement.

$|p_1| = 4298$ objets, $|p_2| = |p_3| = 1$ objet,
 $|p_4| = 200$ objets.

$$s = \frac{4500 \times 1}{5} = 900 \text{ unités de temps.}$$

Etape 1 -

$$A_1 = \{1\}, A_{L_1} = \{2\}, L_1 = 3, A_{L_2} = \{3\},$$

$$L_2 = 4, A_p = A_5 = \{4\}.$$

Nous avons $s = 2$ le nombre des systèmes à résoudre, plus le système d'ordre 0.

900	0	0	0
900	0	0	0
898	1	0	0
900	0	1	0
700	0	0	200

Etape 2 - $|p_1| \times 1 \nmid |p_2| \cdot 1 + |p_3| + |p_4| \times 1$ car $4298 \times 1 > 202 \times 1$.

Donc on calcule d_0 , $d_0 = \min_{i \in A - A_p} \{d^i\}$; $A - A_p = \{1, 2, 3\}$

pour $i=1$:

pour $k=1$ nous avons $d_1^1 = 2$; pour $k=2$ nous avons $d_2^1 = 2$ jours,

donc $d^1 = 2$ jours.

pour $i=2$ et $i=3$ et pour $k=1$ nous avons $d_1^2 = d_1^3 = 5$ jours

donc $d_0 = d^1 = 2$ jours avec l'ordonnancement associé au système d'ordre 0 ;

d'abord p_i , $i \in A^1$ puis p_1

puis p_k , $k \in A_p$, c'est-à-dire p_4 .

donc $d = d_0 = 2$ et $I = 0$, $h = 1$.

Etape 3 -

On commence par calculer d_{A_1} , c'est-à-dire pour $k=1,2$, car $L_1=3$,

$$\sum_{l=1}^k x_{li} \times 1 \leq (d_{A_1} + k-1) \times 3, \text{ d'où } d_{A_1} = 1 \text{ jour}$$

$d_{A_1} \nless d$ donc on calcule $d_{B_{L_1}}$, $B_{L_1} = \{2,3\}$ c'est-à-dire

pour $k=3,4$ car $p=5$;

$$|p_1| \times 1 + \sum_{j \neq i} |p_j| \cdot 1 + \sum_{l=1}^k x_{li} \times 1 \leq (d_{B_{L_1}} + k-1) \times s$$

$$i, j \in B_{L_1}$$

Or pour $i=2$ ou $i=3$ on trouve $d_3^i = 3$ jours, donc $d_{B_{L_1}} = 3$; par conséquent :

$$d_1 = \text{Max} \{d_{A_{L_1}}, d_{B_{L_1}}\} = 3.$$

Puisque $d_h = d_1 > d = d_0$ car $3 > 2$;

comme $h = 1 \leq s-1 = 2-1 = 1$ on compare

$$\sum_{A_{L_1} \cup A_{L_2} \cup A_p} |p_j| \cdot b_j \text{ et } (L_2 - L_1 + 1) \times s \text{ or}$$

$$(|p_2| + |p_3| + |p_4|) \times 1 = 202 \times 1 < (4 - 3 + 1) \times 900 = 1800$$

Donc on arrête la recherche de d et inutile de calculer d_2 vérifiant le système d'ordre 2. D'après l'algorithme (2.3.1) d le nombre de jours de décalage est égal à $d_0 = 2$ avec le système d'ordre 0.

Exemple 4 -

Soit $n=3$ et $p=5$, l'engagement journalier de 2ème traitement est représenté par la matrice suivante où
 $|p_3| = 500$ objets, $|p_2| = |p_1| = 1000$ objets
 et $S = 500$ unités.

Etape 1 -

$A_1 = \{3\}$, $A_{L_1} = \{2\}$ et $L_1 = 2$; $A_{L_2} = \{1\}$ et
 $L_2 = 4$.

$$s = 2$$

Etape 2 -

$\sum_{A_1} |p_j| x_1 < \sum_{A_{L_1} \cup A_{L_2}} |p_j| x_1$ donc $d_1 \leq d_0$

donc on initialise $d_0 = p+1$.

p_1	p_2	p_3
0	0	500
0	500	0
0	500	0
500	0	0
500	0	0

Etape 3 - Nous allons rappeler les résultats rapidement :

avec le système d'ordre 1 on trouve $d_1 = 2$, donc on calcule d_2 vérifiant
 le système d'ordre 2, on trouve $d_2 = 1$ et l'ordonnancement est p_3, p_2

puis p_1 .

Il suffit de regarder le système d'ordre 2 suivant dans ce cas :

$\forall k \in \{1\}$ (car $L_1 = 2$)

$$x_{13} \leq (d_2 + k - 1) \times S = d_2 \times S$$

$\forall k \in \{2, 3\}$ ($L_2 = 4$)

$$|p_3| + \sum_{l=2}^k x_{l2} \leq (d_2 + k - 1) \times S$$

$\forall k \in \{4\}$ ($L_2 = 4$ et $p-1=4$)

$$|p_3| + |p_2| + x_{41} \leq (d_2 + k - 1) \times S$$

Exemple 5 -

On considère le nombre de parts $n=9$ et le nombre de jours de travail de la période $p=10$; soit la matrice suivante, représentant l'engagement journalier de 2ème traitement.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
1		50					41	10	55
2		31	85					34	
3		125	15	55					20
4		50				43		89	
5			68		38	39			
6	35		82		60			9	
7		25	127						
8	35		100		20	40			
9	30		70			39			
10	100			25		45			

$|p_1| = 200$ objets ; $|p_2| = 281$ objets ; $|p_3| = 547$ objets ; $|p_4| = 80$ objets ;

$|p_5| = 118$ objets ; $|p_6| = 206$ objets ; $|p_7| = 41$ objets ; $|p_8| = 142$ objets ;

$|p_9| = 75$ objets.

$S = \left(\sum_{j=1}^9 |p_j| \times 1 : p \right) = (1690 : 10) = 169$ unités de temps par jour de capacité d'engagement moyen pour le 1er traitement.

Etape 1 - $A_1 = \{2, 7, 8, 9\}$; $A_{L_1} = \{3\}$; $L_1 = 2$;

$A_{L_2} = \{4\}$; $L_2 = 3$; $A_{L_3} = \{6\}$; $L_3 = 4$;

$A_{L_4} = \{5\}$; $L_4 = 5$; $A_{L_5} = \{1\}$; $L_5 = 6$.

Le nombre de systèmes à résoudre est $s=5$, plus le système d'ordre 0.

Etape 2 -

$$\sum_{A_1} |p_j| \times 1 = 539 < \sum_{A-A_1} |p_j| \times 1 = 1151 \text{ donc } d_1 \leq d_0 \text{ et } d_0 = p+1, I=0, h=1.$$

Etape 3 - D'abord $h=1 \leq s = 5$,

$$1/ d_{A_1} = \min_{i \in A_1} \{d_1^i\} = d_1^2 = 2, \text{ car } d_1^7 = 4, d_1^8 = 3 \text{ et } d_1^9 = 4.$$

Donc $d_{A_1} = 2$ et l'ordre sur A_1 est $p_j, j \in A_1^2$ puis p_2 .

$$d_{A_1} < d_0 \text{ donc on peut calculer } d_{B_{L_1}} = \min_{i \in \{1,3,4,5,6\}} \{d^i\}$$

$$\text{on trouve } d_{B_{L_1}} = d_2^3 = 7, \text{ car } d_2^1 = 8, d_2^4 = 9, d_2^5 = 9, d_2^6 = 8$$

et l'ordonnancement de B_{L_1} est $p_j, j \in B_{L_1}^3$ puis p_3 ;

$$\text{donc } d_1 = \max \{d_{A_1}, d_{B_{L_1}}\} = 7$$

$$2/ d_1 < d_0 ; h=h+1=2 < s=5 \text{ donc on va calculer } d_{A_{L_1}}$$

$$d_{A_{L_1}} = \min_{i \in A_{L_1}} \{d^i\} \text{ avec } A_{L_1} = \{3\}, \text{ on trouve } d^3 = 3$$

$$\text{donc } d_{A_{L_1}} = d^3 = 3, \text{ comme } d_{A_{L_1}} < d_1 = 7 \text{ on calcule } d_{B_{L_2}}$$

$$\text{avec } B_{L_2} = \{1,4,5,6\}, d_{B_{L_2}} = d_3^6 = d_3^1 = 7$$

L'ordonnancement associé à B_{L_2} est $p_j, j \in B_{L_2}^6$ puis p_6 où $p_j, j \in B_{L_2}^1$ puis p_1 .

$$d_2 = \max \{d_{A_1}, d_{A_{L_1}}, d_{B_{L_2}}\} = \max \{2, 3, 7\} = 7 \text{ jours.}$$

$$3/ d_2 \text{ n'est pas strictement supérieur à } d_1 \text{ donc } h=h+1=3 < s=5$$

$$\text{Donc on calcule } d_{A_{L_2}} \text{ avec } A_{L_2} = \{4\} \text{ on trouve } d^4 = d_3^4 = 5$$

$$\text{Maintenant on calcule } d_{B_{L_3}} \text{ avec } B_{L_3} = \{1,5,6\}, \text{ on trouve } d_{B_{L_3}} = d_4^1 = 6$$

$$\text{car } d_4^6 = d_4^5 = 7$$

$$\text{donc } d_3 = \text{Max } \{d_{A_1}, d_{A_{L_1}}, d_{A_{L_2}}, d_{B_{L_3}}\} = \text{Max } \{2, 3, 5, 6\} = 6$$

4/ $d_3 < d_2$ et $h=h+1=4 < s=5$ donc on calcule

$$d_{A_{L_3}} \text{ avec } A_{L_3} = \{6\}, \text{ on trouve } d^6 = 5$$

$$d_{A_{L_3}} < d_3 \text{ on peut calculer } d_{B_{L_4}} \text{ avec } B_{L_4} = \{1, 5\}$$

$$\text{on trouve } d_{B_{L_4}} = d_5^1 = 5 \text{ car } d_5^5 = 6 \text{ donc}$$

$$d_4 = \text{Max } \{d_{A_1}, d_{A_{L_1}}, d_{A_{L_2}}, d_{A_{L_3}}, d_{B_{L_4}}\} = \{2, 3, 5, 5, 5\} = 5 \text{ jours.}$$

5/ $d_4 < d_3$ et $h=5 < s=5$

$$\text{Calculons } d_{A_{L_4}} \text{ avec } A_{L_4} = \{5\}, \text{ on trouve } d^5 = 5$$

$$\text{donc } d_{A_{L_4}} = 5 \nless d_4 = 5 \text{ on passe au calcul de } d_{B_{L_5}} \text{ avec cette fois}$$

$$B_{L_5} = A_{L_5} = \{1\}, \text{ on trouve } d_6^1 = 5 = d^1$$

donc $d_5 = 5$ puis $h=5$ et $s=5$, fin de l'algorithme.

Donc d nombre de jours de décalage entre les deux traitements est donné par le système d'ordre 5 et l'ordonnancement suivant :

- $p_j, j \in A_1^2$ puis p_2 ,
- p_3 ,
- p_4 ,
- p_6 ,
- p_5 ,
- p_1 .

Exemple 6 -

Soit $n=3$ parts et $p=5$ jours, soit la matrice suivante représentant l'engagement journalier de 2ème traitement.

$$\sum_{A_1} |p_j| \cdot b_j \nless \sum_{A-A_1} |p_j| \cdot b_j$$

Le système d'ordre 0 donne un nombre de jours de décalage $d_0 = 3$ et l'ordonnancement suivant :

$$p_j, j \in A^1 \text{ puis } p_1$$

Le système d'ordre 1 donne $d_1=4 > d_0$ avec l'ordre suivant des parts :

$$p_1 \text{ puis } p_2 \text{ puis } p_3$$

ou

$$p_1 \text{ puis } p_3 \text{ puis } p_2$$

p_1	p_2	p_3
9		
	9	
	1	8
9		
9		

$$\text{Mais } \sum_{A_{L_1} \cup A_{L_2}} |p_j| \times 1 = |p_2| + |p_3| \nless (L_2 - L_1 + 1) \times S$$

$$10 \times 1 + 9 \times 1 \nless (3 \cdot 2 + 1) \times 9$$

$$18 \nless 18$$

Donc on calcule d_2 vérifiant le système d'ordre 2 et $d_2 = 3$ avec l'ordre p_1, p_2 et p_3 .

V - GENERALISATION -

Solution du problème $n|m|1, m_1|D1, (D2, T2, T2.1)|C1.1$ et $C2$ -

On suppose que le 2ème traitement est composé de m_1 chaînes, $m_1 > 1$, m_1 fixé, les m_1 chaînes travaillant en parallèle, avec une seule chaîne de 1er traitement. Le nombre des parts est $n \geq 1$, et le nombre des types est $m \geq 1$. L'ordonnancement de 2ème traitement est fixé. Avec les notations de ce chapitre, soit $X_1, X_2, \dots, X_{m_1}$ les matrices représentant l'engagement journalier de la chaîne 1, 2, ..., m_1 ème chaîne de 2ème traitement.

Algorithme (2.5.1) -

La solution du problème ci-dessus qui consiste à trouver le nombre de jours de décalage minimum entre les deux traitements et l'ordonnancement minimum respectant le principe (1.1) pour le 1er traitement est donnée par :

l'algorithme (2.1) si $n = 1$,

l'algorithme (2.3.2.) si $n > 1$,

en prenant $X = X_1 = X_2 + \dots + X_{m_1}$



CHAPITRE III

ORDONNANCEMENT DE 2ème TRAITEMENT

Résumé -

On considère que l'ordonnancement de 1er traitement est fixé dans ce chapitre. On commence par définir la dépendance entre les deux traitements, puis on déterminera le temps de transition minimum (et le nombre de jours de décalage) pour un ordonnancement commençant par un type donné. On cherchera ensuite l'ordonnancement minimum en coût temps de 2ème traitement. On présentera quelques exemples numériques et on déterminera le programme du lancement le plus économique pour m_1 chaînes travaillant en parallèle ($m_1 > 1$).

I/ RAPPEL : CONTRAINTES ET DEFINITIONS -

L'ordonnancement de 2ème traitement doit respecter les contraintes suivantes, dont on rappelle les plus importantes :

(M2) toutes les chaînes de 1er et 2ème traitement sont dans un seul état : travail sur les objets,

(M4) tout ordonnancement doit respecter l'engagement journalier de chaque chaîne,

(OM1 et OM2) chaque objet de E doit passer par le 1er traitement au plus tard la veille de son engagement dans le 2ème traitement,

(OM3) chaque chaîne traite plusieurs objets à la fois.

De plus la nouvelle contrainte suivante :

(T1) L'ordonnancement de 1er traitement est fixé.

Remarque 3.1. -

- L'ordonnancement de 2ème traitement doit respecter le principe (1.2), travail par type,

- on considère que le temps de transition, si on passe d'un objet d'un type T_j à un objet d'un type T_k est :

$$e_{j,k} = |d_j - d_k| \quad (\text{où } d_j \text{ est le temps d'exécution d'un objet}$$

de type T_j dans le 2ème traitement).

m est le nombre des types. Considérons :

$$d_1 \geq d_2 \geq \dots \geq d_k \geq d_{k+1} \geq \dots \geq d_m$$

en tenant compte du résultat (I.II.2.1.).

$$\text{Le temps de transition est : } \sum_{k=1}^{n-1} e_{k,k+1} = d_1 - d_m$$

Ce temps est obtenu pour un ordre croissant ou décroissant des d_j .

- soit N_i le temps de transition "proprement dit" de 2ème traitement, pour un ordonnancement donné (programme $\mathcal{P}_i$),

- soit z le coût en temps (fixé pour un jour) correspondant au stockage d'objets après le 1er traitement, (par exemple $z=S'$, capacité moyenne journalière de 2ème traitement).

- soit x_i le nombre de jours de décalage entre les deux traitements pour un programme $\mathcal{P}_i$.

Définition (3.1) -

On appelle coût en temps pour un programme du lancement $\mathcal{P}_i$ (respectant les contraintes ci-dessus) :

$$H_i = x_i \times z + N_i$$

Définition (3.2) -

On appelle programme de lancement le plus économique le programme $\mathcal{P}_{i_0}$ tel que :

$H_{i_0} = \text{Min } \{H_i\}$

ou H_i est le coût en temps d'un programme $\mathcal{P}_i$.

II/ DEPENDANCE ENTRE LE 1er ET LE 2ème TRAITEMENT -

Dans le chapitre I nous avons déterminé la relation de dépendance entre le 1er et le 2ème traitement, quand l'ordonnancement de 2ème traitement est fixé. Maintenant c'est l'ordonnancement en parts de 1er traitement (T1) qui est fixé, cette dépendance est étudiée avec une seule chaîne par traitement. La généralisation sera au paragraphe VII).

Soit $T_1, T_2, \dots, T_m$ les types de la période p. A l'aide de la matrice des besoins M , on peut déterminer la matrice $Y = (y_{kj})$
 $1 \leq k \leq p$
 $1 \leq j \leq m$

où y_{kj} est le nombre d'objets de type T_j en différentes parts engagé le k ème jour de 1er traitement.

-Matrice Y - figure (3.1)

T_1	T_2	T_3	T_m		
y_{11}	y_{12}	y_{13}		y_{1m}	1er jour de 1er traitement
y_{21}	y_{22}	y_{23}		y_{2m}	2ème " " "
y_{k1}	y_{k2}	y_{k3}		y_{km}	k ième " " "
y_{p1}	y_{p2}	y_{p3}		y_{pm}	p ième " " "

Rappelons que :

$$1/ \text{ pour } j, 1 \leq j \leq m \quad |T_j| = \sum_{k=1}^p y_{kj}$$

2/ l'engagement journalier moyen de 2ème traitement est :

$$S' = \sum_{j=1}^m |T_j| \cdot d_j : P$$

Si x est le nombre de jours de décalage entre les deux traitements, alors l'engagement journalier de 2ème traitement est respecté (si) (regarder figure I. III. 1.2.).

$$\begin{aligned} \text{pour le 1er jour de 2ème traitement} \quad & \sum_{l=1}^x \sum_{j=1}^m y_{lj} \times d_j \geq S' \times 1 \\ \text{pour le 2ème jour de 2ème traitement} \quad & \sum_{l=1}^{x+1} \left(\sum_{j=1}^m y_{lj} \times d_j \right) \geq S' \times 2 \\ & \vdots \\ \text{pour le } k\text{ème jour de 2ème traitement} \quad & \sum_{l=1}^{x+k-1} \left(\sum_{j=1}^m y_{lj} \times d_j \right) \geq S' \times k \end{aligned}$$

En prenant pour origine de temps le 1er jour de 1er traitement la formule générale ci-dessus s'écrit :

$$\sum_{l=1}^p \sum_{j=1}^m y_{lj} d_j \geq k \times S' \quad \text{pour tout } k \text{ tel que } p-x+1 \leq k \leq p ;$$

On en déduit que l'engagement journalier pour toute la période est respecté si l'engagement journalier des $(p-x)$ premiers jours de 2ème traitement est respecté, c'est-à-dire :

soit $x \in \mathbb{N}^*$ tel que

$$\forall k \in \{1, \dots, p-x\} \quad \sum_{l=1}^{x+k-1} \sum_{j=1}^m y_{lj} \cdot d_j \geq S' \times k \quad (3.1)$$

Soit $D = \{x, x \in \mathbb{N}^*, x \text{ vérifiant } (3.1)\},$

$D \neq \emptyset$, car $x=p$ est une solution évidente.

Conséquence (3.1) -

Pour que l'engagement journalier de 2ème traitement soit respecté, il suffit de prendre le nombre de jours de décalage x_0 :

$$x_0 = \min_{x \in D} \{x\}$$

A/ REECRITURE DU SYSTEME (3.1) -

Pour calculer x dans tous les cas, on va changer l'écriture du système (3.1) :

$$\text{On note par } Q_L = \sum_{j=1}^m y_{Lj} \cdot d_j$$

$$\text{Alors (3.1) s'écrit } \forall k \in \{1, \dots, p-x\} \quad \sum_{l=1}^{x+k-1} Q_L \geq S' \times k$$

$$\text{Soit } h = x+k-1 \quad (\text{h ième jour de 1er traitement})$$

Alors (3.1) s'écrit :

$$\forall k \in \{1, \dots, p-x\} \quad \sum_{l=1}^h Q_L \geq S' \times k \quad (*)$$

Comme $h = x+k-1$ alors $k=h-x+1$ (3.1) s'écrit :

$$\forall k \in \{1, \dots, p-x\} \quad \sum_{l=1}^h Q_L \geq (h-x+1) \cdot S'$$

Dans (*) la valeur maximum de k est $k=p-x$, alors la valeur maximum de h est : $h = x+k-1 = x+p-x-1 = p-1$.

Donc on peut écrire :

$$\forall h \in \{1, \dots, p-1\} \quad \sum_{l=1}^h Q_L \geq (h-x+1) \cdot S' \quad \dots \quad (3.2)$$

B/ EQUIVALENCE ENTRE LES SYSTEMES (3.2) et (3.1) -

Soit x_0 l'entier minimum vérifiant (3.1) :

$$\forall k \in \{1, \dots, p-x_0\} \quad \sum_{l=1}^{x_0+k-1} Q_L \geq S' \times k$$

Soit x l'entier minimum vérifiant (2.2) :

$$\forall h \in \{1, \dots, p-1\} \quad \sum_{l=1}^h Q_L \geq (h-x+1) \cdot S'$$

Montrons d'abord que (3.2) vérifie bien l'engagement journalier de 2ème traitement.

si $\underline{h=x}$, alors (3.2) s'écrit $\sum_{l=1}^x Q_l \geq S' \times 1$

Durant le 1er jour de 2ème traitement l'engagement est respecté.

si $h=x+1$, alors (3.2) s'écrit :

$$\sum_{l=1}^{x+1} Q_l \geq (x+1-x+1).S' = S' \times 2$$

donc l'engagement journalier du 2ème jour est respecté.

Plus généralement,

$$h = x+k-1 \quad \text{et } k \in \{1, \dots, p-x\} ;$$

$$\sum_{l=1}^{x+k-1} Q_l \geq (x+k-1-x+1).S' = S' \times k$$

donc l'engagement journalier du $k^{\text{ième}}$ jour est respecté.

si $h=p-1$, alors

$$\sum_{l=1}^{p-1} Q_l \geq (p-1-x+1).S' = (p-x).S'$$

donc l'engagement journalier de $(p-x)^{\text{ième}}$ jours est respecté ;

donc $x \in D$, par conséquent : $x \geq x_0$

d'après (3.1) nous avons :

$$\forall k \in \{1, \dots, p-x_0\} \quad \sum_{l=1}^{x_0+k-1} Q_l \geq K.S'$$

Alors si on pose $h = x_0+k-1$ avec $k \in \{1, \dots, p-x_0\}$

nous avons $\sum_{l=1}^h Q_l \geq k \times S' = (h-x_0+1).S'$

D'où pour tout $h \in \{1, \dots, p-1\}$

$$\sum_{l=1}^h Q_l \geq (h-x_0+1).S' \geq (h-x+1).S' \quad \text{absurde car } x \text{ est}$$

l'entier minimum vérifiant (3.2), alors $x = x_0$.

C/ SOLUTION DU SYSTEME (3.2) -

pour tout $k \in \{1, \dots, p-1\}$ soit x_k l'entier minimum vérifiant :

$$\sum_{l=1}^k Q_l \geq (k - x_k + 1) \cdot S' ;$$

soit $k_0 \in \{1, \dots, p-1\}$ tel que $x_{k_0} = \max_{k=1}^{p-1} \{x_k\}$

Alors pour tout $k \in \{1, \dots, p-1\}$:

$$\sum_{l=1}^k Q_l \geq (k - x_{k_0} + 1) \cdot S'$$

Théorème (3.1) - Le nombre de jours de décalage minimum entre les deux traitements est :

$$x_{k_0} = \max_{k=1}^{p-1} \left\{ k+1 - \left\lfloor \sum_{l=1}^k Q_l : S' \right\rfloor \right\}$$

Démonstration - Regarder la démonstration du théorème (1.1).

Note - On utilisera toujours le système (3.2) pour vérifier l'engagement journalier de 2ème traitement.

III/ DETERMINATION DU NOMBRE DE JOURS DE DECALAGE ENTRE LES DEUX TRAITEMENTS -Résumé -

Le but est de trouver un système qui respecte l'engagement journalier de 2ème traitement avec le nombre de jours de décalage le plus petit, pour un ordonnancement donné en types.

A) Notation -

Soit $T_1, T_2, \dots, T_m$ les types d'une période p , pour tout $j \in \{1, \dots, m\}$ on note par :

L_j = le dernier jour de traitement pour le type T_j dans le 1er traitement, c'est-à-dire :

$$\forall l \in \{L_j+1, \dots, p\} \quad y_{jl} = 0 \quad \text{ou}$$

y_{lj} = le nombre d'objets en différentes parts de type T_j engagé le $l^{\text{ième}}$ jour de 1er traitement.

Soit γ une permutation de $\{1, \dots, m\}$;

Soit l'ordonnancement suivant :

$$T_{\gamma(1)}, T_{\gamma(2)}, T_{\gamma(3)}, \dots, T_{\gamma(m)} \quad \dots (*)$$

c'est-à-dire dans le 2ème traitement on commence à traiter les objets de type $T_{\gamma(1)}$ (sans interruption), puis les objets de type $T_{\gamma(2)}$, ... puis les objets de type $T_{\gamma(k)}$, puis les objets de type $T_{\gamma(k+1)}, \dots, T_{\gamma(m)}$ ainsi l'ordonnancement (*) est respecté.

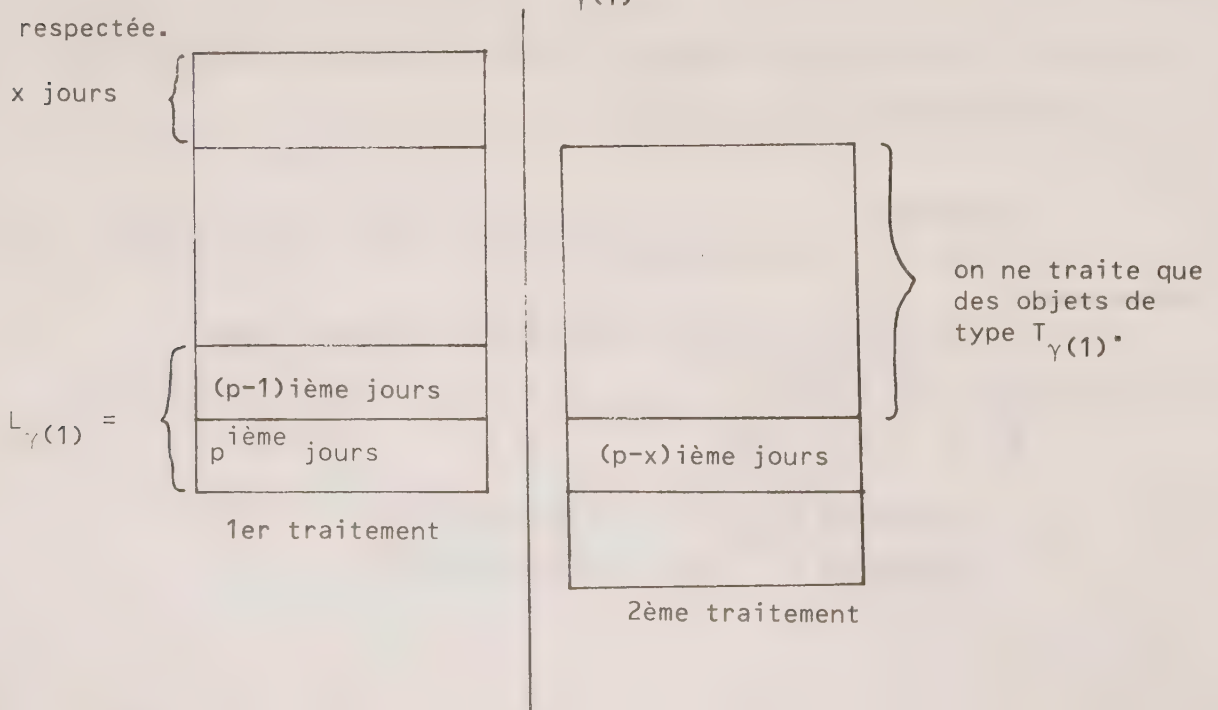
Pour un tel ordonnancement (*) nous allons distinguer deux cas :

B) 1er cas : $L_{\gamma(1)} = p$ (ou $p-1$) -

On suppose que l'ordonnancement $T_{\gamma(1)}, T_{\gamma(2)}, \dots, T_{\gamma(m)}$ est tel que $L_{\gamma(1)} = p$ (ou $p-1$) ; on veut que cet ordonnancement soit respecté de même le principe de travail par type. Alors si x est le nombre de jours de décalage, x doit vérifier :

$$\forall k \in \{1, \dots, p-1\} \quad \sum_{l=1}^k y_{l, \gamma(1)} \cdot d_{\gamma(1)} \geq (k-x+1) \cdot S' \quad \dots (3.3)$$

Le système (3.3) respecte l'engagement journalier de 2ème traitement puisque dans les $(p-x)$ premiers jours de 2ème traitement, on travaille uniquement avec des objets de type $T_{\gamma(1)}$, pour que la contrainte (OM2) soit respectée.



D'après le théorème (3.1) le nombre de jours de décalage minimum est :

$$x = \max_{k=1}^{p-1} \{ k + 1 - \lfloor \left(\sum_{l=1}^k y_{L_Y(l)} \cdot d_Y(l) \right) : s \rfloor \}$$

c) 2ème cas : $L_Y(1) < p-1$ -

On considère que l'ordonnancement $T_Y(1), T_Y(2), \dots, T_Y(m)$ est tel que $L_Y(1) < p-1$, rappelons qu'on cherche à déterminer un système respectant l'ordonnancement avec un nombre de jours de décalage minimum.

a) soit $i_0 \in \{1, \dots, m\}$ tel que :

$$i_0 = \min_{i=2}^m \{ i, \text{ tel que } \left| \begin{array}{l} L_Y(i) > L_Y(1) \text{ et} \\ L_Y(i) \geq p-1 \end{array} \right. \}$$

b) soit $j_1 \in \{2, \dots, i_0-1\}$ tel que :

$$j_1 = \min_{i=2}^{i_0-1} \{ i, L_Y(i) > L_Y(1) \}$$

c) soit $h \geq 2$, on définit :

$$j_h = \min_{i=j_{h-1}+1}^{i_0-1} \{ i \mid L_Y(i) > L_Y(j_{h-1}) \}$$

Soit $s = \text{card} \{ j_1, j_2, \dots, j_h, \dots, j_{s-1}, j_s \}$.

On va s'intéresser aux indices suivants :

$$1 < j_1 < j_2 < \dots < j_h < \dots < j_s < i_0 \quad \text{tel que}$$

$$L_Y(1) < L_Y(j_1) < \dots < L_Y(j_h) < \dots < L_Y(j_s) < L_Y(i_0)$$

Donc l'ordonnancement $T_Y(1), T_Y(2), \dots, T_Y(m)$ s'écrit (avec $i_0 \leq m$) :

$$T_Y(1), \dots, T_Y(j_1), \dots, T_Y(j_2), \dots, T_Y(j_h), \dots, T_Y(j_{h+1}), \dots, T_Y(i_0), \dots, T_Y(m).$$

De plus, on constate que pour $k \in \{j_{h+1}, \dots, j_{h+1}-1\}$ nous avons :

$$L_Y(k) \leq L_Y(j_h).$$

On vérifie que le système (3.4) ci-dessus vérifie :

- les contraintes exprimées dans le 1er paragraphe,
- le principe de travail par type,
- l'ordonnancement.

Système (3.4) -

Soit x l'entier minimum vérifiant :

$$(3.4) \quad \begin{array}{ll} \forall k \in \{1, \dots, L_Y(1)\} & \sum_{l=1}^k y_{L_Y(1)} \cdot d_Y(1) \geq (k-x+1) \cdot S' \\ \forall k \in \{L_Y(1)+1, \dots, L_Y(j_1)\} & \sum_{i=1}^{j_1-1} |T_Y(i)| \cdot d_Y(i) + \sum_{l=1}^k y_{L_Y(j_1)} \cdot d_Y(j_1) \geq (k-x+1) \cdot S' \\ \forall k \in \{L_Y(j_1)+1, \dots, L_Y(j_2)\} & \sum_{i=1}^{j_2-1} |T_Y(i)| \cdot d_Y(i) + \sum_{l=1}^k y_{L_Y(j_2)} \cdot d_Y(j_2) \geq (k-x+1) \cdot S' \\ \vdots & \\ \forall k \in \{L_Y(j_{h-1})+1, \dots, L_Y(j_h)\} & \sum_{i=1}^{j_h-1} |T_Y(i)| \cdot d_Y(i) + \sum_{l=1}^k y_{L_Y(j_h)} \cdot d_Y(j_h) \geq (k-x+1) \cdot S' \\ \vdots & \\ \forall k \in \{L_Y(j_s)+1, \dots, L_Y(i_0)\} & \sum_{i=1}^{i_0-1} |T_Y(i)| \cdot d_Y(i) + \sum_{l=1}^k y_{L_Y(i_0)} \cdot d_Y(i_0) \geq (k-x+1) \cdot S' \end{array}$$

N.B. - Si $L_Y(i_0) = p$ alors pour la dernière inégalité : $L_Y(i_0) = L_Y(i_0) - 1$.

D) Solution du système (3.4) -

Etape 1 -

Pour tout $k \in \{1, \dots, L_Y(1)\}$

Soit $x^k = k+1 - \left\lfloor \sum_{l=1}^k y_{L_Y(1)} \cdot d_Y(1) \right\rfloor : S'$

Soit $k_0 \in \{1, \dots, L_Y(1)\}$ tel que : $x^{k_0} = \max_{k=1}^{L_Y(1)} \{x^k\}$

Etape 2 -

Pour tout $k \in \{L_Y(j_h)+1, \dots, L_Y(j_{h+1})\}$, $h \in \{1, \dots, s\}$

$$\text{Soit } x^k = k+1 - \left\lfloor \left(\sum_{i=1}^{j_{h+1}-1} |T_Y(i)| d_Y(i) + \sum_{l=1}^k y_{L_Y(j_{h+1})} d_Y(j_{h+1}) \right) : S \right\rfloor$$

Soit $l_{j_{h+1}} \in \{L_Y(j_h)+1, \dots, L_Y(j_{h+1})\}$, $h \in \{1, \dots, s\}$

$$\text{Tel que : } x^{l_{j_{h+1}}} = \max_{k=L_Y(j_h)+1}^{L_Y(j_{h+1})} \{ x^k \}$$

Etape 3 -

Pour tout $k \in \{L_Y(j_s)+1, \dots, L_Y(i_o) = p-1\}$

$$\text{Soit } x^k = k+1 - \left\lfloor \left(\sum_{i=1}^{i_o-1} |T_Y(i)| d_Y(i) + \sum_{l=1}^k y_{L_Y(i_s)} d_Y(i_s) \right) : S \right\rfloor$$

Soit $l_{i_o} \in \{L_Y(j_s), \dots, L_Y(i_o)\}$

$$\text{Tel que : } x^{l_{i_o}} = \max_{k=L_Y(j_s)}^{p-1} \{ x^k \}$$

Etape 4 -

L'entier minimum vérifiant le système (3.4) est :

$$x = \max \{ x^{k_o}, x^{l_{j_1}}, x^{l_{j_2}}, \dots, x^{l_{j_s}}, x^{l_{i_o}} \}$$

si on note par $k_o = l_{j_o}$, et $l_{i_o} = l_{j_{s+1}}$ alors $x = \max_{h=0}^{s+1} \{ x^{l_{j_h}} \}$

La démonstration de l'étape 4 est donnée par le théorème (3.1).

D'après l'écriture du système (3.4) et le calcul de x entier minimum vérifiant le système (3.4), on peut annoncer le théorème suivant qui sera très utile ultérieurement.

Théorème 3.2 -

Pour tout ordonnancement en type, commençant par T_j , l'entier minimum de nombre de jours de décalage x entre les deux traitements est supérieur ou égal à x^j ou x^j est l'entier minimum vérifiant :

$$\forall k \in \{1, \dots, L_j\} \quad \sum_{l=1}^k y_{lj} \cdot d_j \geq (l - x^j + 1) \cdot S' \quad (3.**)$$

Démonstration -

Considérons $\gamma(1) = j$ alors :

$$x^j = x \quad \text{si} \quad L_{\gamma(1)} = p \text{ ou } (p-1) \quad \text{système (3.3)}$$

$$x \geq x^j \quad \text{si} \quad L_{\gamma(1)} < p-1 \quad \text{système (3.4)}$$

En effet, si $L_{\gamma(1)} < p-1$ nous avons $x^j = x^{l_j 0}$.

N.B. - On constate que pour les $(n-1)!$ ordonnancements commençant par T_j le nombre de jours de décalage minorant est x^j et le nombre de jours de décalage maximum est x .

IV - ORDONNANCEMENT MINIMUM EN TEMPS DE TRANSITION -

On cherche à définir l'ordonnancement minimum en temps de transition si on veut commencer par un type T_i (i fixé) dans le 2ème traitement. Si $T_1, T_2, \dots, T_m$ sont les types de la période p , on suppose que :

$$d_1 \geq d_2 \geq \dots \geq d_{i-1} \geq d_i \geq d_{i+1} \geq \dots \geq d_m$$

où d_i = le temps de traitement d'un objet de type T_i .

1/ L'ordonnancement commence par T_1 ou T_m -

Si l'ordonnancement en type de 2ème traitement commence par T_1 (respectivement T_m) alors l'ordonnancement de temps de transition minimum consiste à prendre les types dans l'ordre décroissant (resp. croissant) du temps de traitement d'un objet ; c'est-à-dire :

avec T_1 nous avons l'ordonnancement : $T_1, T_2, T_3, \dots, T_i, T_{i+1}, \dots, T_m$;

avec T_m nous avons l'ordonnancement : $T_m, T_{m-1}, \dots, T_i, T_{i-1}, \dots, T_2, T_1$.

Les deux ordonnancements précédents admettent un temps de transition égal à $d_1 - d_m$, (résultat I. II. 2.1).

2/ L'ordonnancement commence par T_i avec $i \neq 1$ et $i \neq m$ -

Alors l'ordonnancement en type commençant par T_i admettant le temps de transition minimum est :

$$a) T_i T_{i-1} \dots T_2 T_1 T_{i+1} T_{i+2} \dots T_{m-1} T_m$$

ou :

$$b) T_i T_{i+1} \dots T_{m-1} T_m T_{i-1} T_{i-2} \dots T_2 T_1$$

Le temps de transition pour l'ordonnancement a) est noté $\Pi_a(i)$:

$$\begin{aligned} \Pi_a(i) &= \sum_{j=1}^{i-1} |d_{j+1} - d_j| + |d_1 - d_{i+1}| + \sum_{j=i+1}^{m-1} |d_j - d_{j+1}| \\ &= d_1 - d_i + d_1 - d_{i+1} + d_{i+1} - d_m \\ &= \underline{2d_1 - d_i - d_m} \end{aligned}$$

Le temps de transition pour l'ordonnancement b) est noté $\Pi_b(i)$:

$$\begin{aligned} \Pi_b(i) &= \sum_{j=i}^{m-1} |d_j - d_{j+1}| + |d_m - d_{i-1}| + \sum_{j=1}^{i-2} |d_{j+1} - d_j| \\ &= d_i - d_m + d_{i-1} - d_m + d_1 - d_{i-1} \\ &= \underline{d_1 + d_i - 2d_m} \end{aligned}$$

Tout ordonnancement en type commençant par T_i (i fixé, $i \neq 1$ et m) différent de l'ordonnancement a) et b) admet un temps de transition supérieur à $\Pi_a(i)$ ou $\Pi_b(i)$.

Démonstration -

Soit τ_{hk} une transposition définie sur $\{1, \dots, m\}$, on considère les trois cas suivants :

- 1) $1 \leq h < k < i$,
- 2) $1 \leq h < i < k \leq m$
- 3) $i < h < k \leq m$

Appliquons τ_{hk} sur l'ordonnancement a tel que h et k vérifient 1) ; on obtient l'ordonnancement suivant noté σ :

$$\sigma = T_i \dots T_{k+1} T_h T_{k-1} \dots T_{h+1} T_k T_{h-1} \dots T_1 T_{i+1} \dots T_m$$

Temps de transition de σ est Π_σ :

$$\Pi_\sigma = (d_{k+1} - d_i) + (d_h - d_{k+1}) + (d_h - d_{k-1}) + (d_{h+1} - d_{k-1}) + (d_{h+1} - d_k) + (d_{h-1} - d_k) +$$

$$(d_1 - d_{h-1}) + (d_1 - d_{i+1}) + (d_{i+1} - d_m)$$

$$= -d_i + 2d_h - 2d_{k-1} + 2d_{h+1} - 2d_k + d_1 + (d_1 - d_{i+1}) + (d_{i+1} - d_m)$$

$$= (d_1 - d_i) + 2(d_h + d_{h+1} - d_k - d_{k+1}) + (d_1 - d_m)$$

$$\text{Comme } d_h \geq d_k \text{ et } d_{h+1} \geq d_{k+1}$$

$$\text{donc } 2(d_h + d_{h+1} - d_k - d_{k+1}) > 0$$

$$\Pi_\sigma = 2(d_h + d_{h+1} - d_k - d_{k+1}) + 2d_1 - d_i - d_m > \Pi_a(i)$$

De même façon, on calcule Π_σ tel que h et k vérifient 2/ et 3/, on en déduit que pour tout h et k différent de i , on a $\Pi_\sigma > \Pi_a(i)$. Donc pour toute permutation σ_i telle que $\sigma_i(i) = i$ nous avons $\Pi_{\sigma_i} > \Pi_a(i)$; il en est de même pour l'ordonnancement b).

3/ Réécriture de l'ordonnancement a) et b) -

Le but de ce changement d'écriture est de faciliter le calcul de nombre de jours de décalage avec le système (3.4) pour tel ordonnancement. Rappelons l'écriture de l'ordonnancement a) :

$$T_i \quad T_{i-1} \quad \dots \quad T_2 \quad T_1 \quad T_{i+1} \quad \dots \quad T_m .$$

Soit σ_i une permutation de $\{1, \dots, i\}$ définie ainsi :
pour tout $j \in \{1, \dots, i\}$ $\sigma_i(j) = i-j+1$.

La restriction de σ_i à $\{i+1, \dots, m\}$ est l'application identique.
(a) devient :

$$T'_1, T'_2, \dots, T'_{i-1}, T'_i, T'_{i+1}, \dots, T'_m .$$

Donc l'ordonnancement (a) est noté (a') :

$$(a') \quad T'_{\sigma_i(1)}, T'_{\sigma_i(2)}, \dots, T'_{\sigma_i(j)}, \dots, T'_{\sigma_i(m)} .$$

De même l'ordonnancement (b) est noté (b') :

$$(b') \quad T'_{\tau_i(m)}, T'_{\tau_i(m-1)}, \dots, T'_{\tau_i(j)}, \dots, T'_{\tau_i(1)} .$$

Avec : $\tau_i(j) = m-j+i$ pour tout $j \in \{m, \dots, i\}$

$$\tau_i(j) = j \quad \text{pour tout } j \in \{i-1, \dots, 1\}$$

On constate qu'il est facile de calculer le nombre de jours de décalage pour l'ordonnancement (a) qui est devenu (a') (avec §.4), pour l'ordonnancement (b) qui est devenu (b'). Le système qui déterminera le nombre de jours de décalage x entre les deux traitements est défini de la même façon que (3.4) en remarquant que :

$$i_0 = \max_{j=1}^{m-1} \{j \text{ tel que } L_{\tau_i(j)} > L_{\tau_i(m)} \text{ et } L_{\tau_i(j)} \geq p-1\}$$

de même :

$$- j_1 = \max_{j=i_0+1}^{m-1} \{j, \text{ tel que } L_{\tau_i(j)} > L_{\tau_i(m)}\}$$

et pour $h > 1$ l'indice j_h est :

$$j_h = \max_{j=i_0+1}^{j_{h-1}} \{j, \text{ tel que } L_{\tau_i}(j) > L_{\tau_i}(j_{h-1})\}$$

avec $s' = \text{card} \{j_1, j_2, \dots, j_h, \dots, j_{s'}\}$

nous avons $m > j_1 > j_2 > \dots > j_h > \dots > j_{s'} > i_0$ de plus

$$L_{\tau_i}(m) < L_{\tau_i}(j_1) < \dots < L_{\tau_i}(j_h) < \dots < L_{\tau_i}(j_{s'}) < L_{\tau_i}(i_0)$$

Soit x l'entier minimum vérifiant (3.5) :

$$\begin{aligned}
 & \forall k \in \{1, \dots, L_{\tau_i}(m)\} \quad \sum_{l=1}^k y_{l, \tau_i}(m) \cdot d_{\tau_i}(m) \geq (k-x+1) \cdot s' \\
 & \forall k \in \{L_{\tau_i}(m)+1, \dots, L_{\tau_i}(j_1)\} \quad \sum_{j=j_1+1}^m |T_{\tau_i}(j)| \cdot d_{\tau_i}(j) + \sum_{l=1}^k y_{l, \tau_i}(j_1) \cdot d_{\tau_i}(j_1) \\
 & \qquad \qquad \qquad \geq (k-x+1) \cdot s' \\
 & \quad \vdots \\
 & \forall k \in \{L_{\tau_i}(j_h)+1, \dots, L_{\tau_i}(j_{h+1})\} \quad \sum_{j=j_{h+1}+1}^m |T_{\tau_i}(j)| \cdot d_{\tau_i}(j) + \sum_{l=1}^k y_{l, \tau_i}(j_{h+1}) \cdot \\
 (3.5) \quad & \qquad \qquad \qquad \cdot d_{\tau_i}(j_{h+1}) \geq (k-x+1) \cdot s' \\
 & \quad \vdots \\
 & \forall k \in \{L_{\tau_i}(j_{s'})+1, \dots, L_{\tau_i}(i_0)\} \quad \sum_{j=i_0+1}^m |T_{\tau_i}(j)| \cdot d_{\tau_i}(j) + \sum_{l=1}^k y_{l, \tau_i}(i_0) \cdot d_{\tau_i}(i_0) \\
 & \qquad \qquad \qquad \geq (k-x+1) \cdot s'
 \end{aligned}$$

V - ORDONNANCEMENT MINIMUM DE 2ème TRAITEMENT -

Nous allons donner la solution du problème $n|m|1,1|(D1,T1),D2|C1.2$ et C2. Il y a n parts, m types, une seule chaîne par traitement. L'ordonnement du 1er traitement est fixe ; le critère (C1.2) minimise le temps de transition de 2ème traitement $\sum_{k=1}^{s-1} e_{j_k, j_{k+1}}$ et le critère (C2) minimise

le nombre de jours de décalage entre les deux traitements. On respecte le principe (1.2) de travail par type et l'engagement journalier des chaînes.

Parmi les ordonnancements, il serait intéressant de trouver celui ou ceux qui donnent le décalage minimum ; s'il y a plusieurs solutions, il est préférable d'en prendre une qui donne le coût de transition minimum. Cette recherche ne semble pas réalisable sans examiner les $m!$ ordonnancements possibles. Pour éviter cela, nous allons examiner le coût et le décalage pour les solutions qui commencent par un type donné. Ensuite, nous pourrions éliminer des solutions qui sont de toute évidence mauvaises.

Nous allons étudier ce problème d'abord pour $m=1,2,3$ et $m > 3$.

A/ $m=1$ (m est le nombre de types) -

Il est clair que la solution du problème $(n|1|1,1|(D1,T1),D2|C1.2$ et C2) est donnée par le système (3.2) donnant le décalage minimum, donc (C2) est satisfait, comme $m=1$, (C1.2) est satisfait puisque le temps de transition (de 2ème traitement) est nul.

Alors l'ordonnement de 2ème traitement est le même que le 1er traitement, avec un nombre de jours de décalage donné par (3.2).

B/ $m=2$ (m est le nombre de types) -

(solution du problème $(n|2|1,1|(D1,T1),D2|C1.2$ et C2)).

Soit T_1 et T_2 les deux types de la période p , soit d_1 (resp. d_2) le temps de traitement d'un objet de type T_1 (resp. T_2). Etant donné que l'on respecte le principe de travail par type, le temps de transition pour un ordonnancement est toujours égal à $|d_1 - d_2|$, par suite l'ordonnement minimum est celui admettant le nombre de jours de décalage le plus petit.

Si $L_j =$ le dernier jour de traitement pour le type T_j dans le 1er traitement (c'est-à-dire $\forall l \in \{L_j+1, \dots, p\} \quad y_{lj} = 0$).

Pour trouver le bon ordonnancement, on distingue deux cas :

1er cas $L_1 \neq L_2$ -

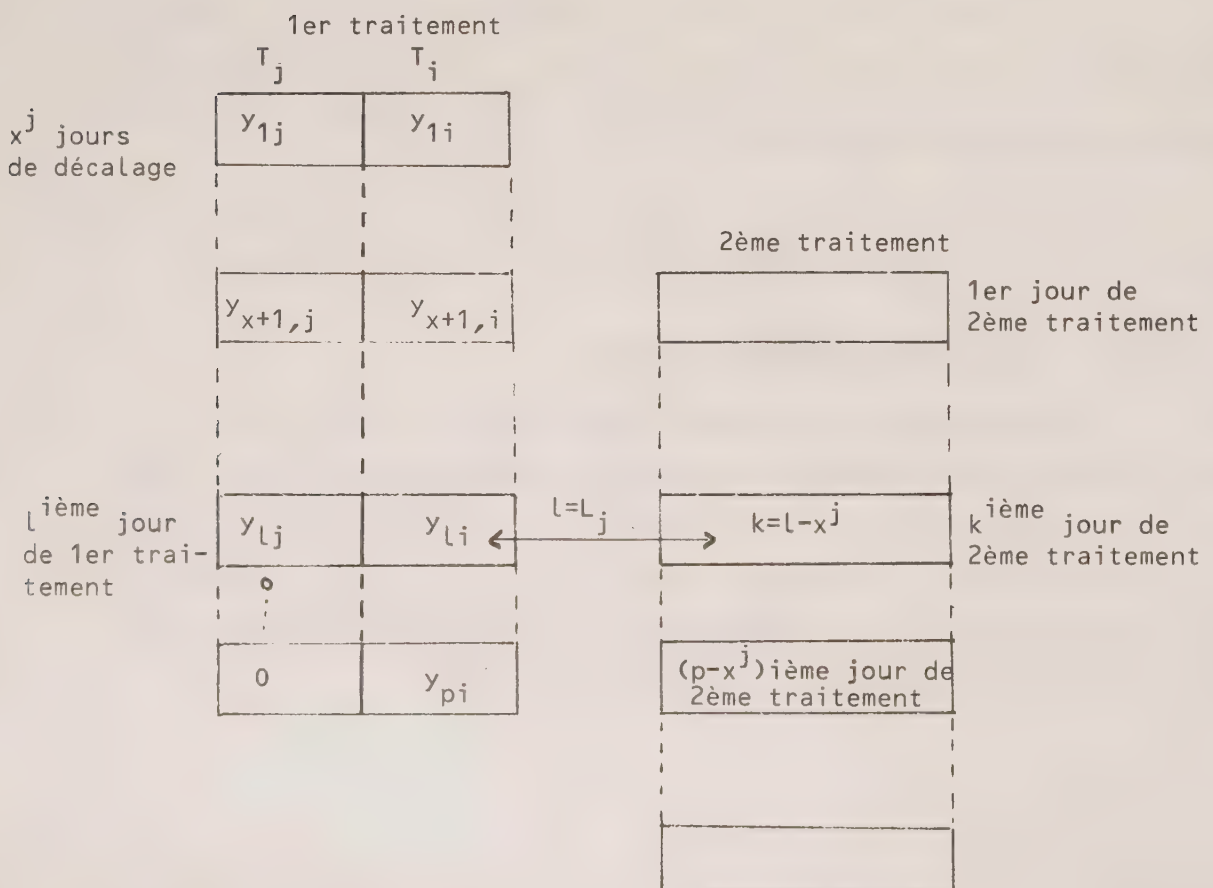
Comme le 1er traitement se termine le $p^{\text{ième}}$ jour, alors nécessairement il existe $i \in \{1, 2\}$ tel que $L_i = p$ (sinon le nombre de jours de travail de 1er traitement serait $< p$).

Soit $j \neq i$, $j \in \{1, 2\}$, rappelons que $L_j < p$, supposons qu'on commence à travailler par les objets de type T_j dans le 2ème traitement, alors le nombre de jours de décalage minimum est calculé en appliquant (3.4) à l'ordre 2 ; ainsi :

soit x^j l'entier minimum vérifiant

$$(3.4.1) \quad \forall l \in \{1, \dots, L_j\} \quad \sum_{k=1}^l y_{kj} \cdot d_j \geq (l - x^j + 1) \cdot S'$$

$$\forall l \in \{L_j+1, \dots, p-1\} \quad |T_j| \cdot d_j + \sum_{k=1}^l y_{ki} \cdot d_i \geq (l - x^j + 1) \cdot S'$$



a) à la date $k^{\text{ième}}$ jour de 2ème traitement, on travaille encore les objets de type T_j (regarder le schéma précédent) puisque dans le 1er traitement, on traite toujours d'objets de type T_j ($k=l-x^j=L_j-x^j$).

b) à la date $(k+1)^{\text{ième}}$ jour de 2ème traitement, nous avons deux possibilités de travailler :

α) les objets de T_j sont tels que l'engagement de $(k+1)^{\text{ième}}$ jour est entièrement assuré et dans ce cas pas de changement de types,

β) à un instant donné dans le $(k+1)^{\text{ième}}$ jour, les objets de T_j sont épuisés, donc on peut engager les objets de type T_i , donc il y a changement de type avec un temps de transition égal $|d_j-d_i|$.

On en déduit que pour α et β .b) on peut toujours considérer le temps de travail pour le $(k+1)^{\text{ième}}$ jour égal à $X - |d_j-d_i|$ où X est le temps de travail "proprement dit" associé à un jour, et dans ces conditions le système (3.4.1) devient :

$$\left| \begin{array}{l} \forall l \in \{1, \dots, L_j\} \quad \sum_{k=1}^L y_{kj} \cdot d_j \geq (l-x^j+1) \cdot X \\ \forall l \in \{L_j+1, \dots, p-1\} \quad |T_j| \cdot d_j + |d_j-d_i| + \sum_{k=1}^L y_{ki} d_i \geq (l-x^j+1) \cdot X \end{array} \right.$$

2ème cas $L_1 = L_2 -$

Les deux types terminent leur traitement le $p^{\text{ième}}$ jour de 1er traitement, dans ce cas : Soit x l'entier minimum vérifiant :

$$(3.3.1) \quad \left| \begin{array}{l} \exists i \in \{1, 2\} \quad \text{tel que :} \\ \forall l \in \{1, \dots, p-1\} \quad \sum_{k=1}^L y_{ki} d_i \geq (l-x+1) \cdot S' \end{array} \right.$$

c'est-à-dire on commence à engager dans le 2ème traitement les objets de type admettant le nombre de jours de décalage le plus petit.

Des deux cas précédents, on en déduit le résultat suivant :

Résultat 3.5.1 - La solution du problème $n|2|1,1|(D1,T1),D2|C1.2$ et $C2$ est déterminée ainsi :

1/ $L_1 = L_2$ La solution est donnée par le système (3.3.1),

2/ $L_1 \neq L_2$ Soit $L_1 < L_2$

soit x^1 l'entier minimum vérifiant (3.4.1)

soit x^2 l'entier minimum vérifiant (3.3.1) pour $i=2$

soit $k \in \{1,2\}$ tel que $x^k = \text{Min} \{x^1, x^2\}$

Alors la solution est donnée par le système admettant comme solution x^k .

C/ Le nombre de type $m=3$ -

Nous allons décrire un algorithme exact qui donnera la solution du problème $(n|3|1,1|(D1,T1),D2|C1.2 \text{ et } C2)$. D'abord nous allons énoncer le théorème suivant qui justifie l'algorithme.

Théorème (3.5.1) -

Soit x (resp. y) l'entier minimum déterminé par le système (3.4) (resp. 3.5) pour l'ordonnancement a (resp. b)).

Soit $\alpha = \min \{x \cdot z + \Pi_a(i) ; y \cdot z + \Pi_b(i) \}$

Alors α est le plus petit coût temps pour tout ordonnancement commençant par le type T_i .

Démonstration -

Il suffit de remarquer que les ordonnancements dans le cas $m=3$ sont uniques.

Algorithme (3.5.1) -

L'ordonnancement en types le plus économique dans le cas $m=3$ est déterminé ainsi :

Etape_1. - pour tout $i \in \{1,2,3\}=A_j$, calculer x^i avec le système (3**) du (théorème 3.2).

Etape 2. - Soit $B = \{i, \text{ tel que } L_i < p-1\}$ Alors

pour $i \in B$, calculer x_a^i avec le système (3.4) pour l'ordonnancement a)
calculer x_b^i avec le système (3.5) pour l'ordonnancement b)

Etape 3. - Soit $i_1 \in A-B$, tel que

$$H_{i_1} = x_{i_1}^{i_1} \cdot z + \text{Min} \{ \Pi_a(i_1), \Pi_b(i_1) \}$$

$$= \text{Min}_{i \in A-B} \{ x_i^i \cdot z + \text{Min} \{ \Pi_a(i), \Pi_b(i) \} \}$$

Soit $i_2 \in B$, tel que $H_{i_2} = \text{Min}_{i \in B} \{ x_o^i \cdot z + \Pi_a(i) ; x_b^i \cdot z + \Pi_b(i) \}$

Soit $i_o \in \{i_1, i_2\}$ tel que $H_{i_o} = \text{Min} \{ H_{i_1}, H_{i_2} \}$

Alors l'ordonnancement minimum (programme de lancement le plus économique) est celui commençant par T_{i_o} .

Exemple Numérique où $(m=3)$ -

On considère que le nombre de types est $m=3$, avec $p=5$ jours de travail. Soit la matrice Y représentant l'engagement journalier de 1er traitement ; de plus le temps de traitement pour un objet de type: T_1 est $d_1=3$

T_1	T_2	T_3
	900	
		900
		900
900		
900		

T_2 est $d_2=2$

T_3 est $d_3=1$

Pour $T_1 \rightarrow L_1 = p$

$T_2 \rightarrow L_2 = 1$

$T_3 \rightarrow L_3 = 3$

$$S' = \sum_{i=1}^3 |T_i| \cdot d_i : p = (1800 \times 3 + 900 \times 2 + 1800 \times 1) : 5$$

= 1800 unités de temps.

Etape 1 - pour $i=1,2,3$ nous avons x^i qui est l'entier minimum vérifiant

$$\forall l \in \{1, \dots, L_i\} \quad \sum_{k=1}^l y_{ki} d_i \geq (l-x^i+1) \cdot S'$$

(si $L_i=p$ alors $L_i=L_i-1$)

d'où $x^1 = 4$ jours atteint pour $l=3$ et $l=4$

$$x^2 = 1 = 1+1 - \lfloor \frac{900 \times 2 + 1800}{1800} \rfloor = 1+1-1=1$$

$x^3 = 3$ atteint pour $l=2$ et $l=3$

Etape 2 - $B = \{2,3\}$ car $L_2 = 1$ et $L_3 = 3$ sont $< p-1 = 4$.

pour T_2 (a) : $T_2 T_1 T_3$

(b) : $T_2 T_3 T_1$

L'ordonnancement (a') est $T'_{\sigma_2(1)} T'_{\sigma_2(2)} T'_{\sigma_2(3)}$, $\sigma_i(j) = i-j+1$, $j \in \{1, \dots, i\}$

$L_{\sigma_2(1)} = 1$ et $L_{\sigma_2(2)} = p$ donc $i_0=2$, les indices $j_1, \dots, j_s$ n'existent pas. On applique le système (3.4).

$$\forall l \in \{1, \dots, L_{\sigma_2(1)}\} = \{1\} \quad y_{1\sigma_2(1)} \cdot d_{\sigma_2(1)} \geq S' \cdot x(1+1-x)$$

$$\forall l \in \{L_{\sigma_2(1)}, \dots, L_{\sigma_2(i_0)}-1\} \quad |T_{\sigma_2(1)}| \cdot d_{\sigma_2(1)} + \sum_{k=1}^l y_{k\sigma_2(i_0)} d_{\sigma_2(i_0)} \geq S' \cdot x(l+1-x)$$

$$\text{pour } l=1 \quad x^{\sigma_2(1)} = x^2 = 1$$

$$l=2 \quad x_2 = 2+1 - \lfloor \frac{900 \times 2 + 0 \times 3}{1800} \rfloor = 2$$

$$l=3 \quad x_3 = 3+1 - \lfloor \frac{900 \times 2 + 0 \times 3}{1800} \rfloor = 3$$

$$l=4 \quad x_4 = x^{\sigma_2(i_0)} = 4+1 - \lfloor \frac{900 \times 2 + 900 \times 3}{1800} \rfloor = 2$$

donc $x = \text{Max} \{1, 2, 3, 2\} = 3$ jours

Même chose pour (b), on trouve $y=2$ jours avec (3.5).

Alors l'ordonnancement minimum commençant par T_2 est l'ordonnancement (b) car :

$$- \Pi_a(2) = 2d_1 - d_2 - d_3 = 2 \times 3 - 2 - 1 = 3$$

$$- \Pi_b(2) = d_1 + d_2 - 2d_3 = 3 + 2 - 2 \times 1 = 3$$

donc $2 \times z + \Pi_b(2) < 3 \times z + \Pi_a(2)$ avec $\Pi_a(2) = \Pi_b(2)$.

Pour T_3 , nous avons l'ordonnancement $T_3 T_2 T_1$ de coût $\Pi(3) = 2$ comme $x^3 = 3$ et $L_3 = 3$. Inutile d'appliquer le système (3.5) dans ce cas puisque le 1er traitement se termine à la fin du 2ème jour de 2ème traitement.

Coût d'ordonnancement commençant par T_3 est $3 \times z + 2$

Etape 3 -

$$A - B = \{1, 2, 3\} - \{2, 3\} = \{1\}$$

Coût de programme du lancement commençant par T_1 est :

$$4 \times z + d_1 - d_3 = 4 \times z + 2.$$

L'ordonnancement de coût minimum pour $i \in B$ est l'ordonnancement **(b)** commençant par T_2 , c'est-à-dire : $T_2 T_3 T_1$ de coût $2 \times z + 3$; ceci en prenant ($z = S'$).

D/ Le nombre de types est $m \geq 4$ (m fixé) -

Avant de proposer une solution au problème $(n|m|1,1|(D1,T1),D2|C1.2$ et C2), nous allons chercher à définir l'ordonnancement minimum en coût temps pour un ordonnancement commençant par un type T_j donné, avec les $j \in \{1, \dots, m\}$

mêmes notations de ce chapitre.

Soit l'ordonnancement suivant : $T_{\gamma(1)}, T_{\gamma(2)}, \dots, T_{\gamma(m)}$, avec

$$T_{\gamma(1)} = T_j.$$

$$1/ \underline{L_{\gamma(1)}} = L_j \geq p-1 -$$

Le nombre de jours de décalage x^j pour un tel ordonnancement est unique pour les $(m-1)!$ ordonnancements commençant par T_j . Le nombre de jours de décalage est déterminé par le système (3.3), il est minimum, de plus nous avons montré que pour tout type T_j l'ordonnancement minimum en temps de transition est (a) ou (b).

Dans ces conditions on peut énoncer le théorème suivant :

Théorème 3.5.2 -

Soit $X = \{j, \text{ tel que } L_j \geq p-1\}$

pour tout $j \in X$, soit $H_j = x^j \cdot z + \text{Min} \{ \pi_a(j), \pi_b(j) \}$

Alors H_j est le plus petit coût en temps parmi les $(m-1)!$ ordonnancement commençant par T_j , tel que $L_j \geq p-1$.

Démonstration -

Remarquons que x vérifiant le système (3.*.*) (du théorème 3.2) est égal à x^j vérifiant le système (3.3). De plus pour tout $x' \in \mathbb{N}^*$, $x' < x^j$, x' ne vérifie pas l'engagement journalier de 2ème traitement.

$$2/ \underline{L_{\gamma(1)}} = L_j < p-1 -$$

On suppose que l'ordonnancement commençant par le type T_j est tel que $L_j < p-1$;

Soit x^j l'entier minimum vérifiant le système (3.*.*) du (th. 3.2)

Soit x l'entier minimum vérifiant le système (3.4)

Soit y l'entier minimum vérifiant le système (3.5)

Rappelons toujours que l'ordonnancement minimum en temps de transition, commençant par le type T_j , est l'ordonnancement a) ou b) ;

Si x est le nombre de jours de décalage pour a), si y est le nombre de jours de décalage pour b).

*) On peut distinguer deux cas :

- si $x^j = x$ alors le coût en temps pour l'ordonnancement a) est minimum,
- si $x^j = y$ alors le coût en temps pour l'ordonnancement b) est minimum.

car dans ces deux cas nous avons le nombre de jours de décalage minimum avec le temps de transition minimum.

**) On suppose que $x^j < x$ ou $(x^j < y)$ dans ce cas nous allons chercher à réduire le nombre de jours de décalage de telle sorte que le nouvel ordonnancement commençant par T_j ait un temps de transition minimum (pour le nouvel ordre des types). Auparavant donnons la définition suivante, qui donnée

pour l'ordonnancement a), sera valable pour l'ordonnancement b) :

Définition (3.5.1) -

Soit la suite $y^k \leq y^{k-1} \leq \dots \leq y^2 \leq y^1 = x$ tel que
pour $h \in \{1, \dots, k\}$, $y^h \in [x^j, x] \cap \mathbb{N}^*$ et y^h vérifie l'engagement
journalier de 2ème traitement ainsi que le principe de travail par type.
On suppose que $\forall y \in [x^j, y^k] \cap \mathbb{N}^*$, y ne vérifie pas l'engagement journalier
de 2ème traitement ou bien le principe de travail par type (on peut avoir
 $[x^j, y^k]_{k \geq 2} = \emptyset$ dans certains cas).

Soit $h_0 \in \{1, \dots, k\}$ tel que :

$$y^{h_0} \times z + \Pi_{y^{h_0}}(j) = \min_{h=1}^k \{ y^h \times z + \Pi_{y^h}(j) \}$$

On appelle ordonnancement minimum en coût temps l'ordonnancement réalisé
avec y^{h_0} jours de décalage et un temps de transition $\Pi_{y^{h_0}}(j)$.

Réduction du nombre de jours de décalage (Heuristique d'amélioration) -

Nous cherchons à réduire x pour l'ordonnancement (a). La méthode
sera valable pour l'ordonnancement (b) (dans certains cas la réduction est
impossible).

L'ordonnancement a) : $T_i \ T_{i-1} \ \dots \ T_i \ T_{i+1} \ \dots \ T_m =$

$$T'_{\sigma_i}(1) \ T'_{\sigma_i}(2) \ \dots \ T'_{\sigma_i}(m) =$$

$$T'_{\sigma_i}(1) \ \dots \ T'_{\sigma_i}(j_1) \ \dots \ T'_{\sigma_i}(j_0) \ \dots \ T'_{\sigma_i}(i_0) \ \dots \ T'_{\sigma_i}(m)$$

Alors x l'entier minimum déterminé par (3.4) est : (avec $i_0 \leq m$)

$$x = \max \{x^i, x^{l_{j1}}, x^{l_{j2}}, \dots, x^{l_{js}}, x^{l_{i_0}}\}$$

Rappelons que x^i est obtenu sur l'ensemble $\{1, \dots, L_{\sigma_i}(1)\}$

pour $h \in \{1, \dots, s+1\}$ ($x^{L_{j_{s+1}}} = x^{L_{j_0}}$)

$x^{L_{j_h}}$ est obtenu sur l'ensemble $\{L_{\sigma_i(j_{h-1})}+1, \dots, L_{\sigma_i(j_h)}\}$

avec $L_{\sigma_i(j_0)} = L_{\sigma_i(1)}$.

Soit $x = x^{L_{j_{h_0}}}$ avec $x^{L_{j_{h_0}}}$ obtenu sur l'ensemble

$\{L_{\sigma_i(j_{h_0-1})}+1, \dots, L_{\sigma_i(j_{h_0})}\}$ $h_0 \in \{1, \dots, s+1\}$.

L'idée est de diminuer d'abord x sur cet ensemble.

- L'ordonnancement (a) s'écrit :

$T'_{\sigma_i(1)} \dots T'_{\sigma_i(j_{h_0-1})} T'_{\sigma_i(j_{h_0-1}+1)} \dots T'_{\sigma_i(j_{h_0})} T'_{\sigma_i(j_{h_0}+1)} \dots T'_{\sigma_i(m)}$

On constate d'abord que $\forall j \in \{1, \dots, j_{h_0}-1\}$ $L_{\sigma_i(j)} < L_{\sigma_i(j_{h_0})}$

$\forall k \in \{1, j_1, \dots, j_{h_0}-1\}$ $x^{L_{j_k}} < x$

- Diminuer x revient à changer l'ordre de l'ordonnancement, c'est-à-dire faire reculer $T'_{\sigma_i(j_{h_0})}$ le minimum de positions possibles, par

conséquent le nouvel ordonnancement (a₁) admet dans ces conditions un temps de transition minimum pour le nouvel ordre.

Démonstration -

On distingue trois cas :

1/ $i = j_{h_0}$

2/ $i < j_{h_0}$

3/ $j_{h_0} < i$

Montrons que le temps de transition (pour le 1/ ($i=j_{h_0}$)) est minimum en faisant reculer $T'_{\sigma_i(j_{h_0})}$:

$\sigma_i(i) = i - i + 1 = 1$, donc $h, k > j_{h_0}$ avec $h < k$ $d_{\sigma_i(h)} \geq d_{\sigma_i(k)}$

car $d_1 \geq d_2 \geq \dots \geq d_m$, la démonstration est finie si l'inégalité suivante est vraie.

$$|d_{\sigma_i}(j_{h_0}-1) - d_{\sigma_i}(h)| + |d_{\sigma_i}(h) - d_{\sigma_i}(j_{h_0})| \leq$$

$$|d_{\sigma_i}(j_{h_0}-1) - d_{\sigma_i}(k)| + |d_{\sigma_i}(k) - d_{\sigma_i}(j_{h_0})|$$

$$\cdot d_{\sigma_i}(j_{h_0}-1) \geq d_{\sigma_i}(h) \geq d_{\sigma_i}(k)$$

$$\cdot -d_{\sigma_i}(k) \geq -d_{\sigma_i}(h) \text{ donc } d_{\sigma_i}(j_{h_0}-1) - d_{\sigma_i}(k) \geq d_{\sigma_i}(j_{h_0}-1) - d_{\sigma_i}(h) \dots (\alpha)$$

de même $|d_{\sigma_i}(h) - d_{\sigma_i}(j_{h_0})| = d_{\sigma_i}(j_{h_0}) - d_{\sigma_i}(h)$

$$|d_{\sigma_i}(k) - d_{\sigma_i}(j_{h_0})| = d_{\sigma_i}(j_{h_0}) - d_{\sigma_i}(k) \geq d_{\sigma_i}(j_{h_0}) - d_{\sigma_i}(h) \quad (\beta)$$

avec (α) et (β) l'inégalité est vérifiée. La même méthode est applicable pour 2/ et 3/.

Prenons maintenant $h=j_{h_0}+1$ et regardons la fin de l'ordonnancement a) qui devient (a_1) , sachant que le début de l'ordonnancement reste fixe pour $j < j_{h_0}-1$;

$$(a_1) \dots T'_{\sigma_i}(j_{h_0}-1) T'_{\sigma_i}(j_{h_0}+1) T'_{\sigma_i}(j_{h_0}) T'_{\sigma_i}(j_{h_0}+2) \dots T'_{\sigma_i}(m)$$

ou bien :

$$(a_1) \dots T'_{\sigma_i}(j_{h_0}-1) T'_{\sigma_i}(j_{h_0}+2) T'_{\sigma_i}(j_{h_0}) T'_{\sigma_i}(j_{h_0}+1) T'_{\sigma_i}(j_{h_0}+3) \dots T'_{\sigma_i}(m)$$

Pour avancer un type donné on prend en compte :

- son dernier jour de traitement dans le 1er traitement,
- son temps d'exécution en 2ème traitement de ses objets ayant terminé leur traitement dans le 1er traitement.

Etape Réduction -

DEBUT $M := j_{h_0} + 1 ;$ TANTQUE $M \leq m$ FAIREDEBUTSI $L_{\sigma_i}(m) \leq L_{\sigma_i}(j_{h_0}-1)$ ALORS DEBUT

- 1- Ajouter $|T_{\sigma_i}(M)| \cdot d_{\sigma_i}(M)$ au membre gauche de l'inégalité du système (3.4) définie par l'ensemble $\{L_{\sigma_i}(j_{h_0}-1)+1, \dots, L_{\sigma_i}(j_{h_0})\}$
- 2- Calculer $x_{j_{h_0}}$.

③- Calculer x_{a_1} (l'entier minimum vérifiant (3.4))

si $x_{a_1} > x^i$ Alors $M := M+1 ;$

sinon $M := m+1 ;$ (en effet $x_{a_1} \leq x^i$ fin de réduction)

FINSINON DEBUT

Si $|T_{\sigma_i}(M)| \cdot d_{\sigma_i}(M) \leq \sum_{k=1}^{L_{\sigma_i}(M)} y_{k, \sigma_i}(j_{h_0}) \cdot d_{\sigma_i}(j_{h_0})$ ALORS DEBUT

si $L_{\sigma_i}(M) < L_{\sigma_i}(j_{h_0})$ ALORS DEBUT

①- Redéfinir les indices à partir de $j_{k_0}-1$, pour le système (3.4) ;

③-

FINSINON $M := M+1 ;$ FINSINON DEBUT

①-

③-

FIN ;FIN ;FIN ;FIN ;

Avec cette heuristique nous avons obtenu des réductions optimums sur tous les exemples essayés.

Algorithme (3.5.2) -

Solution du problème $(n|m|1,1|(D1,T1),D2|C1.2 \text{ et } C2)$
 $m \geq 4$

Rappelons qu'on cherche l'ordonnancement minimum en coût temps pour le 2ème traitement.

Etape 1. -

Pour tout $i \in A = \{1,2,\dots,m\}$, calculer x^i entier minimum vérifiant (3.*.*) du th. (3.2).

Etape 2. -

Soit $C = \{i \in A, \text{ tel que } L_i \geq p-1\}$

soit $i_0 \in C$ tel que

$$H_{i_0} = x^{i_0} \cdot z + \text{Min} \{ \Pi_a(i_0), \Pi_b(i_0) \} = \text{Min}_{i \in C} \{ x^i \cdot z + \text{Min} \{ \Pi_a(i), \Pi_b(i) \} \}$$

Etape 3. -

Soit $B = A - C = \{i \in A, L_i < p-1\}$

soit $i_1 \in B$ tel que $x^{i_1} = \text{Min}_{i \in B} \{x^i\}$,

soit $i_2 \in B$ tel que $x^{i_2} = \text{Min}_{i \in B} \{x^i, x^i > x^{i_1}\}$

soit $i_s \in B$ tel que $x^{i_s} = \text{Max}_{i \in B} \{x^i\}$

ainsi on vient de définir les ensembles :

$$B_1 = \{i, x^i = x^{i_1}\} ; \text{ pour } h \text{ tel que } 2 \leq h \leq s, B_h = \{i, x^i = x^{i_h}\}$$

Si $H_{i_0} \leq (x^{i_1} \cdot z + d_1 - d_n)$ Alors fin de la recherche d'ordonnancement minimum

SINON $H := H_{i_0}$;

Etape 4. -

$i := 1$

TANT QUE $i \leq s$ FAIRE

DEBUTpour tout $j \in B_i$ FAIRE DEBUT

- 1- Calculer x_a^j avec (3.4) pour l'ordonnancement (a)
- (*) 2- si $x_a^j > x^i$ Alors réduire x_a^j
 soit $x_{a_1}^j, \dots, x_{a_r}^j$ les entiers trouvés par (3.4) après réduction
- 3- soit $H_{a_{r_0}}^j = \min_{k=1}^r \{x_{a_k}^j \cdot z + \pi_{a_k}(j)\}$

(*) par l'ordonnancement (b) avec (3.5).

FIN ;

- soit $j_i \in B_i$ tel que $H^{j_i} = \min_{j \in B_i} \{H_{a_{r_0}}^j, H_{b_{r_0}}^j\}$

- soit $H = \min \{H, H^{j_i}\}$

soit $k \in B_{i+1}$

si $H \leq (x^k \cdot z + d_1 - d_n)$ Alors $i := i+1$; (car fin de recherche d'ordonnancement minimum)

SINON $i := i+1$;FINVI - EXEMPLES NUMERIQUES -

Exemple 1. - Soit $n=5$ le nombre des types, soit $p=5$ le nombre de jours de la période. On considère l'ordonnancement fixe (de 1er traitement) représenté par la matrice suivante

T_1	T_2	T_3	T_4	T_5
40	30	30	100	
50				150
	160	20	20	
200				
		200		

3 2,5 2 1,5 1 ← temps de traitement d'un objet d_i

$\forall i \in \{1, \dots, 5\}$ Calculons $|T_i| \cdot d_i$

$$|T_1| \cdot d_1 = 290 \times 3 = 870 ;$$

$$|T_2| \cdot d_2 = 190 \times 2,5 = 475 ;$$

$$|T_3| \cdot d_3 = 250 \times 2 = 500 ;$$

$$|T_4| \cdot d_4 = 120 \times 1,5 = 180 ;$$

$$|T_5| \cdot d_5 = 150 \times 1 = 150 ;$$

$$S' = \sum_{i=1}^5 |T_i| \cdot d_i : 5 = 2175 : 5 = 435$$

$S' = 435$ unités.

$\forall i \in \{1, \dots, 5\}$ Calculons x^i entier minimum vérifiant

$$\forall l \in \{1, \dots, L_i\} \quad \sum_{j=1}^L y_{ji} \cdot d_i \geq (l - x^i + 1) S' \quad \text{si } L_i = p \text{ alors } L_i = p-1$$

$$x^i = \max_{l=1}^{L_i} \left\{ 1 + l - \left\lfloor \frac{\sum_{j=1}^L y_{ji} \cdot d_i}{S'} \right\rfloor \right\} = \max_{l=1}^{L_i} \{ x_l^i \}$$

Etape 1. -

Calculons x^1 pour T_1 , nous avons $L_1 = 4$ d'où

pour $l=1$ $x_1^1 = 1+1 - \lfloor 40 \times 3 : 435 \rfloor = 2$

$l=2$ $x_2^1 = 1+2 - \lfloor (40+50) \times 3 : 435 \rfloor = 3$

$l=3$ $x_3^1 = 1+3 - \lfloor (40+50+0) \times 3 : 435 \rfloor = 4$

$l=4$ $x_4^1 = 1+4 - \lfloor (40+50+0+200) \times 3 : 435 \rfloor = 4$

$$\text{d'où } \underline{x^1 = 4}$$

Calculons x^2 pour T_2 , nous avons $L_2 = 3$

$l=1$ $x_1^2 = 1+1 - \lfloor 30 \times 2,5 : 435 \rfloor = 2$

$l=2$ $x_2^2 = 1+2 - \lfloor (30+0) \cdot 2,5 : 435 \rfloor = 3$

$l=3$ $x_3^2 = 1+3 - \lfloor (30+0+160) \cdot 2,5 : 435 \rfloor = 3$

$$\text{d'où } \underline{x^2 = 3}$$

Calculons x^3 pour T_3 avec $L_3 = 5$ donc $L_3 = 4$

nous avons : $l=1 \Rightarrow x_1^3 = 2$

$l=2 \Rightarrow x_2^3 = 3$

$l=3 \Rightarrow x_3^3 = 4$

$l=4 \Rightarrow x_4^3 = 5$

d'où $x^3 = 5$

De même façon on calcule x^4 pour T_4 avec $L_4 = 3$ d'où $x^4 = 4$

x^5 pour T_5 avec $L_5 = 2$ d'où $x^5 = 3$

Etape 2. -

$$C = \{1, 3\} = \{i, L_i \geq p-1\}$$

Ordonnancement pour T_1 et $T_1 T_2 T_3 T_4 T_5$

avec un coût $= 4 \times z + d_1 - d_5 = 4 \times z + 2$

Considérons $z = S' = 435$ $H_1 = 4 \times 435 + 2 = 1742$

Ordonnancement pour T_3 (a) $T_3 T_2 T_1 T_4 T_5$ $\Pi_a(3) = 2d_1 - d_3 - d_5 = 6 - 2 - 1 = 3$

(b) $T_3 T_4 T_5 T_2 T_1$ $\Pi_b(3) = d_1 + d_3 - 2d_5 = 3 + 2 - 1 = 4$

$$H_3 = 5 \times 435 + 3 = 2178$$

d'où $H_{i_0} = H_1 = 1742$

Etape 3. -

$$B = \{i, L_i < p-1\} = \{2, 4, 5\}, \quad x^2 = x^3 = 3, \quad x^4 = 4$$

$$B_1 = \{2, 5\} \quad B_2 = \{4\}$$

$$x^2 \times z + d_1 - d_5 < H_{i_0} \quad \text{donc Etape 4.}$$

Etape 4. -

pour $i=2$ (a) $T_2 T_1 T_3 T_4 T_5$ $L_1 = 4$

donc $j_1 = j_{s+1} = i_0 = 1, \quad s=0$

(3.4) s'écrit : soit x_4^2 l'entier minimum vérifiant :

$$\forall l \in \{1, 2, 3\} \quad \sum_{k=1}^l y_{k2} d_2 \geq (1 + p \cdot x_1^2) \cdot s' \quad (3.41)$$

$$\forall l \in \{4\} \quad |T_2| \cdot d_2 + \sum_{k=1}^4 y_{k1} \cdot d_1 \geq (1 + 4 \cdot x_1^2) \cdot s' \quad (3.42)$$

(3.41) est vérifié pour $x^2 = 3$ maintenant par (3.42)

$$x_4^{l_1} = 1 + 4 - \lfloor (|T_2| \cdot d_2 + |T_1| \cdot d_1) : s' \rfloor = 1 + 4 - \lfloor (475 + 870) : 435 \rfloor = 1 + 4 - 3 = 2$$

Donc $x_a^2 = x^2 = 3$, coût de l'ordonnancement (a) est

$$3 \times 435 + 2d_1 - d_2 - d_6 = 3 \times 435 + 6 - 2,5 - 1 = 3 \times 425 + 2,5 = 1307,5$$

(b) $T_2 T_3 T_4 T_5 T_1$ $L_3 = 5$ $j_1 = j_{s+1} = i_0 = 3 ; s=0$

(3.5) s'écrit soit x_b^2 l'entier minimum vérifiant :

$$(3.5.1) \quad \forall l \in \{1, 2, 3\} \quad \sum_{k=1}^l y_{k2} d_2 \geq (1 + l \cdot x_b^2) \cdot s'$$

$$(3.5.2) \quad \forall l \in \{4\} \quad |T_2| \cdot d_2 + \sum_{k=1}^4 y_{k3} d_3 \geq (1 + 4 \cdot x_b^2) \cdot s'$$

(3.5.1) est vérifié par $x^2 = 3$,

$$x_4^{l_3} = 1 + 4 - \lfloor (475 + 100) : s' \rfloor = 1 + 4 - 1 = 4$$

donc $x_a^2 = 4$

Réduction de x_b^2 .

(b₁) $T_2 T_4 T_3 T_5 T_1$, $L_4 = 3 < L_3 = 4$
 $L_4 \leq L_2$

(3.5.1) reste fixé, donc (3.5.2) :

$$\forall l \in \{4\} \quad |T_2| \cdot d_2 + |T_4| \cdot d_4 + \sum_{k=1}^4 y_{k3} \cdot d_3 \geq (1+4-x_{b_1}^2) \cdot S'$$

$$x_4^3 = 1 + 4 - |(475 + 180 + 100) : S| = 1+4-1 = 4$$

$$x_{b_1}^2 = 4$$

Réduction de $x_{b_1}^2$

$$(b_2) \quad T_2 \ T_4 \ T_5 \ T_3 \ T_1 \quad L_5 < L_4 = L_2$$

(3.5.1) reste fixe.

$$(3.5.2) \quad \forall l \in \{4\} \quad |T_2| \cdot d_2 + |T_4| \cdot d_4 + |T_5| \cdot d_5 + \sum_{k=1}^4 y_{k3} \cdot d_3 \geq (1+4-x_{b_2}^2) \cdot S'$$

$$x_4^3 = 1+4 - \lfloor (475+180+150+100) : S' \rfloor = 5-2 = 3$$

$$x_{b_2}^2 = x^2 = 3 \quad \text{donc } x^2 \text{ est atteint} \quad \text{coût en temps de transition}$$

$$= d_2 - d_5 + d_3 - d_5 + d_1 - d_3 = d_1 + d_2 - 2d_5 = 3 + 2,5 - 2 = 3,5$$

$$\Pi_{b_2}(2) = 3 \times 435 + 3,5 = 1308,5$$

Ordonnancement minimum commençant par T_2 est (a) :

$$T_2 \ T_1 \ T_3 \ T_4 \ T_5 \text{ de coût } 1307,5$$

Ordonnancement commençant par T_5

$$T_5 \ T_4 \ T_3 \ T_2 \ T_1 \quad L_5=2, \ L_4=3, \ L_3=5$$

$$\text{donc } j_1 = 4 ; j_{s+1} = i_0 = 3, s=1$$

(3.5) s'écrit soit y l'entier minimum vérifiant :

$$\forall l \in \{1,2\} \quad \sum_{k=1}^l y_{k5} \cdot d_5 \geq (l+l-y) \cdot S' \quad (3.5.1)$$

$$\forall l \in \{3\} \quad |T_5| \cdot d_5 + \sum_{k=1}^3 y_{k4} \cdot d_4 \geq (1+3-y) \cdot S' \quad (3.5.2)$$

$$\forall l \in \{4\} \quad |T_5| \cdot d_5 + |T_4| \cdot d_4 + \sum_{k=1}^4 y_{k3} \cdot d_3 \geq (1+4-y) \cdot S' \quad (3.5.3)$$

(3.5.1) est vérifié par $x^5 = 3$

$$(3.5.2) \quad x_3^{L_4} = 1+3 - \lfloor (150+180):435 \rfloor = 4$$

$$(3.5.3) \quad x_4^{L_3} = 1+4 - \lfloor (150+180+100):435 \rfloor = 5$$

$$\underline{y = 5}$$

d'abord réduction pour (3.5.3) c'est-à-dire sur $\{L_4+1, \dots, L_3-1\}$

Le nouvel ordonnancement est :

$T_5 \ T_4 \ T_2 \ T_3 \ T_1$ $L_2 \leq L_4$, donc (3.5.1) et (3.5.2) sont fixes.

(3.5.3) devient :

$$|T_5| \cdot d_5 + |T_4| \cdot d_4 + |T_2| \cdot d_2 + \sum_{k=1}^4 y_{k3} \cdot d_3 \geq (1+4-y') \cdot S'$$

$$x_4^{L_3} = 1+4 - \lfloor (150+180+475+100):S' \rfloor = 5-2 = 3$$

donc $y' = 4 > x^5$, le Max se trouve sur l'intervalle $\{L_5+1, \dots, L_4\}$

Le nouvel ordonnancement $T_5 \ T_2 \ T_4 \ T_3 \ T_1$, $L_2 > L_5$ et $L_2 \leq L_4$ et

$|T_2| \cdot d_2 > |T_4| \cdot d_4$. Redéfinition de $j_1, \dots, j_s$ et i_0 . Donc $j_1=2$, $i_0=3$

$$(3.5.2) \quad \forall l \in \{3\} \quad |T_5| \cdot d_5 + \sum_{k=1}^3 y_{k2} \cdot d_2 \geq (1+3-y'') \cdot S'$$

$$(3.5.3) \quad \forall l \in \{4\} \quad |T_5| \cdot d_5 + |T_2| \cdot d_2 + |T_4| \cdot d_4 + \sum_{k=1}^4 y_{k3} \cdot d_3 \geq (1+4-y'') \cdot S' ,$$

pour (3.5.2) nous avons $x_3^{L_2} = 1+3 - \lfloor (180+475):435 \rfloor = 3$

$$(3.5.3) \quad x_4^{L_3} = 1+4 - \lfloor (180+475+180+100):435 \rfloor = 3$$

donc x^5 est atteint avec l'ordonnancement $T_5 \ T_2 \ T_4 \ T_3 \ T_1$ coût en temps de transition est : $d_2 - d_5 + d_2 - d_4 + d_3 - d_4 + d_1 - d_3 = d_1 + 2d_2 - 2d_4 - d_5$

$$= d_1 - d_5 + 2(d_2 - d_4) = 2 + 2 = 4$$

$$H_5 = 3 \times z + 4$$

$$= 3 \times 435 + 4 = 1309$$

Ordonnancement minimum pour $i \in B_1$ est

$$T_2 \ T_1 \ T_3 \ T_4 \ T_5 \text{ coût } 1307,5 = H_2^1$$

$$\text{Min } \{ H_2^{i_0}, H_2^1 \} = H_2^1$$

$$\text{de plus } H_2^1 < x^4 \times z + d_1 - d_5$$

donc ordonnancement minimum est $T_2 \ T_1 \ T_3 \ T_4 \ T_5$

FIN de l'algorithme.

Exemple 2. - On considère que le nombre des types $n=5$ et le nombre de jours de la période $p=6$.

T_1	T_2	T_3	T_4	T_5	
20	20	20	20	20	
100					
	100				
		100			
			100		
				100	
6	5	4	3	2	$\leftarrow d_i$

$$S' = \sum_{i=1}^n |T_i| \cdot d_i : p = \sum_{i=1}^n (120 \times d_i) : 6 = 2400 : 6 = 400$$

$$|T_1| \cdot d_1 = 720, |T_2| \cdot d_2 = 600, |T_3| \cdot d_3 = 480, |T_4| \cdot d_4 = 360, |T_5| \cdot d_5 = 240$$

$$L_1 = 2, L_2 = 3, L_3 = 4, L_4 = 5, L_5 = 6.$$

Nous allons considérer $z = S'$ par exemple.

Etape 1. -

$\forall i \in \{1, 2, 3, 4, 5\}$ Calculons x^i vérifiant (3.***)

$$x^i = \max_{k=1}^{L_i} \left\{ 1 + k - \left\lfloor \sum_{l=1}^k y_{li} \cdot d_l : S' \right\rfloor \right\}$$

$$i=1 \Rightarrow \underline{x^1 = 2} ; \quad i=2 \Rightarrow \underline{x^2 = 3} ; \quad i=3 \Rightarrow \underline{x^3 = 4}$$

$$\text{pour } i=4 \text{ et } i=5 \text{ nous avons } \underline{x^4 = 6} \quad \text{et} \quad \underline{x^5 = 6}$$

Etape 2. -

$$C = \{i, i \in A, L_i \geq p-1\} = \{4, 5\} \quad T_4 \text{ et } T_5$$

$$T_5 \text{ ordonnancement min } T_5 T_4 T_3 T_2 T_1 \quad \Pi(5) = d_1 - d_5 = 6 - 2 = 4$$

$$H^5 = 6 \times S' + 4 = 6 \times 400 + 4 = 2404$$

T_4 ordonnancement minimum

$$(a) \quad T_4 T_3 T_2 T_1 T_5 \quad \Pi_a(4) = 2d_1 - d_4 - d_3 = 12 - 3 - 2 = 7$$

$$(b) \quad T_4 T_5 T_3 T_2 T_1 \quad \Pi_b(4) = d_1 + d_4 - 2d_5 = 6 + 3 - 4 = 5$$

$$H^4 = 6 \times S' + \Pi_b(4) = 6 \times 400 + 5 = 2405$$

$$\underline{i_o = 5} \quad \text{car } H_{i_o} = 2404$$

Etape 3. -

$$B = \{1, 2, 3\} = \{i \in A, b_i < p-1\}$$

$$\text{Nous avons } x^1 = 2, \quad x^2 = 3, \quad x^3 = 4$$

$$\text{d'où } B_1 = \{1\} \quad B_2 = \{2\} \quad B_3 = \{3\}$$

$$\text{ici nous avons } H_{i_o} > x^1 \times z + d_1 - d_4 = 2 \times S' + 4$$

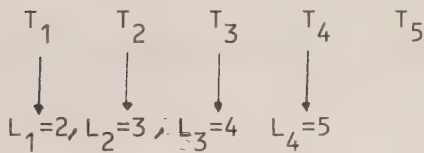
Donc étape 4.

Etape 4. -

$i = 1$ pour tout $j \in B_1$ calculer x_a^j et x_b^j et les réduire

$B_1 = \{1\}$ ici $(a) = (b)$

$T_1 T_2 T_3 T_4 T_5$ nous allons appliquer le système (3.4).



donc $j_1=3, j_2 = j_s = 4, i_0 = j_{s+1} = 5$.

(3.4)

$$\forall l \in \{1,2\} \quad \sum_{k=1}^l y_{k1} \cdot d_1 \geq (1+l-x_a^1) \cdot S' \quad (3.4.1)$$

$$\forall l \in \{3\} \quad |T_1| \cdot d_1 + \sum_{k=1}^l y_{k2} \cdot d_2 \geq (1+l-x_a^1) \cdot S' \quad (3.4.2)$$

$$\forall l \in \{4\} \quad |T_1| \cdot d_1 + |T_2| \cdot d_2 + \sum_{k=1}^l y_{k3} \cdot d_3 \geq (1+l-x_a^1) \cdot S' \quad (3.4.3)$$

$$\forall l \in \{5\} \quad |T_1| \cdot d_1 + |T_2| \cdot d_2 + |T_3| \cdot d_3 + \sum y_{k4} \cdot d_4 \geq (1+l-x_a^1) S' \quad (3.4.4)$$

Le Max pour 3.4.1 est $x_a^1 = 2$,

$$l=3 \quad x_a^{l_2} = 1+3 - \lfloor (|T_1| \cdot d_1 + |T_2| \cdot d_2) : S' \rfloor = 1+3 - \lfloor (720+600) : 400 \rfloor = 2$$

$$l=4 \quad x_a^{l_3} = 1+4 - \lfloor (|T_1| \cdot d_1 + |T_2| \cdot d_2 + |T_3| \cdot d_3) : S' \rfloor = 1+4 - \lfloor (720+600+480) : 400 \rfloor = 1$$

$$l=5 \quad x_a^{l_4} = 1+5 - \lfloor (720+600+480+360) : 400 \rfloor = 1$$

On constate ici que $x_a^1 = x_a^1 = 2$ Rien à réduire.

L'ordonnancement minimum est $T_1 T_2 T_3 T_4 T_5$ de coût $2 \times S' + 4 = 804$

$$H = \text{Min} \{2404, 804\} = 804$$

Soit $k \in B_{1+1=2}$ donc $k = 2$

Nous avons maintenant $H = 804 < x^2 \times S' + d_1 - d_4 = 3 \times 400 + 4 = 1204$
donc fin de recherche de l'ordonnancement minimum.

Exemple 3. - $n = 6, \quad p=5$

	T_6	T_5	T_4	T_3	T_2	T_1
1					100	100
2				100	100	
3	100	100				
4		100	100			
5			100	100		
$d_i \rightarrow$	1	2	3	4	5	6

$$|T_1| \cdot d_1 = 6 \times 100 = 600, \quad |T_2| \cdot d_2 = 1000, \quad |T_3| \cdot d_3 = 800, \quad |T_4| \cdot d_4 = 600,$$

$$|T_5| \cdot d_5 = 400, \quad |T_6| \cdot d_6 = 100.$$

$$D'où \quad S' = (600 + 1000 + 800 + 600 + 400 + 100) : 5 = 3500 : 5 = 700$$

Etape 1. -

$$\text{pour } i=1 \text{ nous avons } \underline{x^1 = 2}$$

$$\text{pour } i=2 \text{ nous avons } \underline{x^2 = 2}$$

$$\text{pour } i=3 \text{ nous avons } \underline{x^3 = 5}$$

$$i=4 \text{ nous avons } \underline{x^4 = 5}$$

$$i=5 \text{ nous avons } \underline{x^5 = 5}$$

$$\text{pour } i=6 \text{ nous avons } \underline{x^6 = 4}$$

Etape 2. -

$$C = \{3, 4, 5\} = \{i, L_i \geq p-1\}$$

$$T_3 \quad (a) \quad T_3 \quad T_2 \quad T_1 \quad T_4 \quad T_5 \quad T_6 \quad \Pi_a(3) = 2d_1 - d_3 - d_6 = 12 - 4 - 1 = 7$$

$$(b) \quad T_3 \quad T_4 \quad T_5 \quad T_6 \quad T_2 \quad T_1 \quad \Pi_b(3) = d_1 + d_3 - 2d_6 = 6 + 4 - 2 = 8$$

$$T_4 \rightarrow \Pi_a(4) = 2d_1 - d_4 - d_6 = 12 - 3 - 1 = 8, \quad \Pi_b(4) = 6 + 3 - 2 = 7$$

$$T_5 \quad \Pi_a(5) = 2d_1 - d_5 - d_6 = 12 - 2 - 1 = 9$$

$$\Pi_b(5) = d_1 + d_5 - d_6 = 6 + 2 - 1 = 7$$

Nous avons 3 ordonnancements minimums

$$T_3 \ T_2 \ T_1 \ T_4 \ T_5 \ T_6 \quad \text{coût } 5 \times z + 7 \quad (\text{ordonnement (a)})$$

$$T_4 \ T_5 \ T_6 \ T_3 \ T_2 \ T_1 \quad \text{coût } 5 \times z + 7 \quad (\text{ordonnement (b)})$$

$$T_5 \ T_6 \ T_4 \ T_3 \ T_2 \ T_1 \quad \text{coût } 5 \times z + 7 \quad (\text{ordonnement (b)})$$

Nous considérons $z = S'$

$$H = 5 \times 700 + 7 = 3507$$

Etape 3 et Etape 4. -

$$B = \{1, 2, 6\}$$

$$B_1 = \{1, 2\}, \quad B_2 = \{6\}.$$

Pour $i=1$ nous avons (a) = (b) = $T_1 \quad T_2 \quad T_3 \quad T_4 \quad T_5 \quad T_6$
 $\downarrow \quad \downarrow$
 $j_1=2 \quad i_0=3 \quad \text{car } L_3=p$

(3.4) s'écrit :

$$\forall l \in \{1\} \quad \sum_{k=1}^L y_{k1} \cdot d_1 \geq (1+l-x_a^1) \cdot S' \quad (3.4.1)$$

$$\forall l \in \{2\} \quad |T_1| \cdot d_1 + \sum_{k=1}^L y_{k2} \cdot d_2 \geq (1+l-x_a^1) \cdot S' \quad (3.4.2)$$

$$\forall l \in \{3, 4\} \quad |T_1| \cdot d_1 + |T_2| \cdot d_2 + \sum_{k=1}^L y_{k3} \cdot d_3 \geq (1+l-x_a^1) \cdot S' \quad (3.4.3)$$

$$l=1 \quad x^{l1} = x^1 = 2,$$

$$l=2 \quad x^{l2} = 1+2 - \lfloor (600+1000):700 \rfloor = 1$$

$$l=3 \quad x^{l3} = 1+3 - \lfloor (600+1000+400):700 \rfloor = 2$$

$$l=4 \quad x^{l4} = 1+4 - \lfloor (600+1000+400+0):700 \rfloor = 3$$

$$\text{D'où } x_a^1 = 3 > x_a^1 \quad \text{réduction de } x_a^1.$$

Etape de réduction

$$(a) \quad T_1 \ T_2 \ T_3 \ T_4 \ T_5 \ T_6 \ , \quad j_{h_0} = i_0 = 3$$

$$(M=4) \quad L_4 > L_2 \quad 4$$

$$\text{de plus } |T_4| \cdot d_4 < \sum_{k=1}^4 y_{k3} d_3 \ , \quad L_4 = L_3 \ (=p-1)$$

donc $M = M+1 = 5$

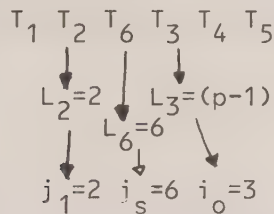
$$\text{même chose pour } T_5 \quad |T_5| \cdot d_5 < \sum_{k=1}^4 y_{k3} \cdot d_3 \ , \quad L_5 = L_3 \ (p-1)$$

donc $M = 6$

$$L_6 > L_2 \quad \text{mais}$$

$$|T_6| \cdot d_6 < \sum_{k=1}^3 y_{k3} d_3 \quad \text{mais } L_6 < L_3$$

Donc le nouvel ordonnancement est :



(3.4.1) et (3.4.2) restent fixes.

$$(3.4.3) \quad \forall l \in \{3\} \quad |T_1| \cdot d_1 + |T_2| \cdot d_2 + |T_6| \cdot d_6 \geq (1+3-x_a^1) \cdot S'$$

$$(3.4.4) \quad \forall l \in \{4\} \quad |T_1| \cdot d_1 + |T_2| \cdot d_2 + |T_6| \cdot d_6 + \sum_{k=1}^4 y_{k3} \cdot d_3 \geq (1+4-x_a^1) \cdot S'$$

$$x_3^{L_6} = 1+3 - |(600+1000+100):700| = 2$$

$$x_4^{L_3} = 1+4 - |(600+1000+100+400):700| = 1+4-3 = 2$$

Donc x^1 est atteint $x^1 = 2$ avec un temps de transition

$$\begin{aligned}
 d_1 - d_6 + d_3 - d_6 + d_3 - d_5 &= d_1 - d_6 + 2d_3 - d_5 \\
 &= 5+6 = 11
 \end{aligned}$$

Le coût de l'ordonnancement commençant par T_1 est : $2 \times S' + 11 = 1411$.

pour T_2 (a) $T_2 \quad T_3 \quad T_4 \quad T_5 \quad T_6 \quad T_1$
 $\downarrow \quad \downarrow$
 $L_2=2 \quad L_3=5$

Calculons pour $l \in \{3,4\}$

$$|T_2| \cdot d_2 + \sum_{k=1}^l y_{k3} \cdot d_3 \geq (1+l-x_a^2) \cdot s'$$

$$l=3 \quad x_3^3 = 1+3 - \lfloor (1000+400):700 \rfloor = 2$$

$$l=4 \quad x_4^3 = 1+4 - \lfloor (1000+400):700 \rfloor = 3$$

$$\text{donc } x_a^2 = 3 > x^2$$

Etape de réduction -

Comme pour la réduction de l'ordonnancement commençant par T_1 , T_4 et T_5 ne bouge pas.

(a₁) $T_2 \quad T_6 \quad T_3 \quad T_4 \quad T_5 \quad T_1$
pour $l=3$ nous avons $x_3^6 = 3$
 $l=4 \quad x_4^3 = 3$

$x_{a_1}^2 = 3$, le Max existe sur deux intervalles. On prend en compte le dernier c'est-à-dire pour $l=4$. Même remarque que précédemment T_4 , T_5 ne bouge pas.

(a₂) $T_2 \quad T_6 \quad T_1 \quad T_3 \quad T_4 \quad T_5$ le Max pour $l=4$ est $x_4^3 = 2$,
de nouveau $x_{a_2}^2 = 3$ pour $l=3$.

(a₃) $T_2 \quad T_1 \quad T_6 \quad T_3 \quad T_4 \quad T_5$ ainsi x^2 est atteint $x_{a_3}^2 = 2$.

Le temps de transition est $d_1 - d_2 + d_1 - d_6 + d_3 - d_6 + d_3 - d_5 = 2d_1 + 2d_3 - d_2 - d_5 - d_6$
 $= 2 \times 6 + 2 \times 4 - 5 - 2 - 1$
 $= 12$

(b) $T_2 \quad T_1 \quad T_3 \quad T_4 \quad T_5 \quad T_6$

même réduction comme si l'ordonnancement commençait par T_1

(b₁) $T_2 \quad T_1 \quad T_6 \quad T_3 \quad T_4 \quad T_5$

x^2 est atteint, le coût est le même que (a_3) :

$$\Pi_b(2) = 12$$

$$H_{b_1}^2 = H_{a_3}^2$$

$$H_{a_3}^2 = 2 \times S' + 12$$

$$H = \min \{H_{a_1}^1, H_{a_3}^2\} = H_{a_1}^1 = 2 \times S' + 11$$

$$H_a^1 < x^6 \times S' + d_1 - d_6 = 4 \times S' + 5 \quad \text{fin de l'algorithme.}$$

Donc l'ordonnancement minimum est $T_1 T_2 T_6 T_3 T_4 T_5$.

VII - GENERALISATION : SOLUTION DU PROBLEME

$$(n|m|1, m_1|(D1, T1), (D2, T2.1)|C1.2 \text{ et } C2)$$

Avec les notations et les contraintes de ce chapitre, on considère de plus la contrainte (T 2.1) que les $m_1 > 1$ machines de 2ème traitement travaillent en parallèle, chaque machine a ses types spécifiques.

En appliquant l'algorithme (3.5.2) à chaque chaîne on obtient :

Le coût en temps	$x_1 \times z + \Pi_1$	pour la 1ère chaîne,
"	"	"
	$x_2 \times z + \Pi_2$	2ème
	$\vdots$	"
Le coût en temps	$x_k \times z + \Pi_k$	kième
	$\vdots$	"
"	"	"
	$x_{m_1} \times z + \Pi_{m_1}$	m_1 ième chaîne

Rappelons que :

x_k = le nombre de jours de décalage pour la kième chaîne

Π_k = le temps de transition pour l'ordonnancement déterminé

z = le coût en temps pour un jour de stockage d'objet après le 1er traitement.

A priori le coût en temps pour la période est :

$$\sum_{k=1}^{m_1} (x_k \times z + \Pi_k)$$

mais les m_1 chaînes commencent à la même date donc le coût en temps est :

$$\left(\max_{k=1}^{m_1} \{ x_k \} \right) \times z + \sum_{k=1}^{m_1} \Pi_k$$

Conséquence 1 -

Si pour tout $(k, k') \in \{1, \dots, m_1\} \times \{1, \dots, m_1\}$, $k \neq k'$

nous avons $x_k = x_{k'}$. Alors dans ce cas le programme du lancement le plus économique de coût égal à :

$$x_k \times z + \sum_{k=1}^{m_1} \Pi_k.$$

mais en général nous n'avons pas $x_1 = x_2 = \dots = x_k = \dots = x_{m_1}$.

D'où l'idée de l'algorithme suivant.

Algorithme 3.7.1. -

La solution du problème $(n|m|1, m_1|(D1, T1), (D2, T2.1)|C1.2 \text{ et } C2)$ est déterminée ainsi :

Etape 1. -

Pour tout $k \in \{1, \dots, m_1\}$, soit $T_k = \{T_{1_k}, T_{2_k}, \dots, T_{m_k}\}$ l'ensemble des types pour la $k^{\text{ième}}$ chaîne de 2ème traitement.

Calculer le nombre des jours de décalage minimum pour chaque type T_{i_k} ($1 \leq i_k \leq m_k$) de l'ensemble T_k , avec le système (3.*.*) du théorème (3.2).

Soit $x_{T_{i_k}}$ le nombre de décalage trouver pour le type T_{i_k} ;

$$\text{soit } x'_{T_{i_k}} = \min_{i=1}^{m_k} \{ x_{T_{i_k}} \}$$

Soit $k_0 \in \{1, \dots, m_1\}$ tel que $x'_{T_{i_k}} = \max_{k=1}^{m_1} \{ x'_{T_{i_k}} \}$

Etape 2. -

Appliquer l'algorithme (3.5.2) à la chaîne d'indice k_0 , soit y le nombre de jours de décalage trouvés.

Etape 3. -

pour tout $k \in \{1, \dots, m_1\}$ et $k \neq k_0$

Soit $T'_k = \{T_{i_k}, x_{T_{i_k}} \leq y\}$

Soit $\Pi_{T_{i_k}}$ le temps de transition de l'ordonnancement commençant par $T_{i_k} \in T'_k$ vérifiant l'engagement journalier de $k^{\text{ième}}$ chaîne de 2ème traitement.

Soit $\Pi'_{T_{i_k}} = \min_{T_{i_k} \in T'_k} \{\Pi_{T_{i_k}}\}$ avec y'_k jours de décalage.

Alors l'ordonnancement minimum associé à la $k^{\text{ième}}$ chaîne est celui commençant par T_{i_k} et admettant comme temps de transition $\Pi'_{T_{i_k}}$.

Etape 4. -

Soit $y'_{k_1} = \max_{k=1}^{m_1} \{y'_k\}$ (on note par $y'_{k_0} = y$)

TANT QUE $y'_{k_1} > y$ FAIRE

DEBUT

$k_0 = k_1$

- Etape 2.

- Etape 3.

- Etape 4.

FIN ;

Remarque 3.7. -

Dans l'étape 3 nous pensons aux ordonnancements (a) et (b) définis dans le paragraphe IV de ce chapitre.

CHAPITRE IV

COMPARAISON AVEC MACHINE FLOW-SHOP PROBLEME

Le début de ce chapitre est destiné à présenter une solution pour les deux problèmes $(n|1|1,m_1|D1,(D2.T2.1)|C1 \text{ et } C2 ; 1|m|1,m_1|D1,(D2.T2.1)|C1 \text{ et } C2)$ sachant que ni l'ordonnancement de 1er, ni l'ordonnancement de 2ème traitement n'est fixé, le but est de trouver ces deux ordonnancements en minimisant le temps de transition et le nombre de jours de décalage ; cette étude nous conduit à la comparaison des problèmes flow-shop pour 2, 3, 4 et 5 machines.

I - SOLUTION DU PROBLEME : $1|m|1,m_1|D1,(D2,T2.1)|C1 \text{ et } C2$ -

A/ Solution du problème $1|m|1,1|D1,D2|C1 \text{ et } C2$ -

(1 part, m type, une seule chaîne par traitement, les mêmes contraintes habituelles, minimiser le temps de transition et le nombre de jours de décalage).

Rappelons qu'on cherche à déterminer pour les deux traitements l'ordonnancement minimum en coût temps.

Le nombre de parts est $n=1$, par conséquent le temps de transition pour le 1er traitement est nul (il n'y a pas de changement en parts) il reste à trouver le temps de transition minimum pour le 2ème traitement et le nombre

de jours de décalage minimum entre les deux traitements, d'où l'algorithme suivant.

Algorithme 4.1. -

Soit $T = \{T_1, T_2, \dots, T_m\}$ ensemble des types, soient T_k et T_j deux éléments de T , tel que $k \neq j$; Alors le type T_k précède le type T_j dans le 2ème traitement en respectant le principe (1.2) travail par type ; (resp. dans le 1er traitement).

Si $d_k > d_j$ (où d_k = temps d'exécution d'un objet de type T_k)

(en cas d'égalité des d_j , on pourra par exemple prendre l'ordre croissant ou décroissant) des quantités d'objets dans les types.

Théorème 4.1. -

L'algorithme 4.1. :

- réalise le temps de transition minimum,
- d'autre part l'ordonnancement est réalisable pour un nombre de jours de décalage $d = 1$.

Démonstration -

Le temps de transition est minimum d'après le résultat I.II.2.1). Montrons maintenant que l'ordonnancement est réalisable avec un jour de décalage, c'est-à-dire il suffit de montrer que l'engagement journalier de 2ème traitement est respecté.

Dans le 2ème traitement, les types sont pris dans l'ordre décroissant des d_j , alors le nombre d'objets traités le premier jour est inférieur ou égal au nombre des objets traités le 2ème jour, de même le nombre d'objets traités le 2ème jour est inférieur ou égal à celui du 3ème jour, d'une manière générale. Le nombre d'objets traités le k ème jour de 2ème traitement est inférieur ou égal à celui du $(k+1)$ ème jour, $1 \leq k \leq p-1$; avec les mêmes notations du chapitre I.

La matrice $X = (x_{k1})_{1 \leq k \leq p}$ représente l'engagement journalier de 2ème

traitement en fonction des parts (ici une seule part).

On en déduit que :

$$x_{11} \leq x_{21} \leq \dots \leq x_{k1} \leq x_{k+1,1} \leq \dots \leq x_{p1}$$

l'engagement journalier moyen de 1er traitement est :

$$S = |p_1| \cdot b_1 : p = \sum_{k=1}^p x_{k1} \cdot b_1 : p \quad \dots \quad (\alpha)$$

$$\text{Soit } Q_k = x_{k1} \cdot b_1 \text{ alors } Q_1 \leq Q_2 \leq \dots \leq Q_p \quad \dots \quad (\beta)$$

Soit d le nombre de jours de décalage. Nous allons montrer que :

$$\forall k \in \{1, \dots, p-d\} \quad \sum_{l=1}^k Q_l \leq (d+k-1) \cdot S \quad (1.3.2)$$

Regardons si le système est vérifié pour $d=1$, c'est-à-dire :

$$\forall k \in \{1, \dots, p-1\} \quad \sum_{l=1}^k Q_l \leq (1+k-1) \cdot S = k \cdot S$$

$$\text{Nous allons montrer que } \forall k \in \{1, \dots, p\} \quad \sum_{l=1}^k Q_l \leq k \cdot S$$

Soit k fixé, $k \in \{1, \dots, p\}$

$$1/ \text{ Si } Q_k \leq S \text{ alors } \sum_{l=1}^k Q_l \leq k \cdot S \text{ car } Q_1 \leq Q_2 \leq \dots \leq Q_k$$

$$2/ \text{ Si } Q_k > S \text{ supposons que } \sum_{l=1}^k Q_l > k \cdot S \quad (1)$$

par hypothèse (β) nous avons $S < Q_k \leq Q_{k+1} \leq \dots \leq Q_p$ donc :

$$\sum_{l=k+1}^p Q_l > (p-k) \cdot S \quad \dots \quad (2)$$

Faisons la somme des inégalités (1) et (2),

$$\sum_{l=1}^k Q_l + \sum_{l=k+1}^p Q_l > k \cdot S + (p-k) \cdot S = p \cdot S ; \text{ c'est-à-dire}$$

$$\sum_{l=1}^p Q_l > p \cdot S \text{ absurde d'après } (\alpha) ; \text{ donc } \sum_{l=1}^k Q_l \leq k \cdot S$$

Donc le décalage entre les deux traitements est un jour, et c'est le minimum.

B/ Solution du problème 1 | m | 1, m₁ | D1, (D2, T2.1) | C1 et C2 -

Nous généralisons le résultat de A/ dans le cas de plusieurs chaînes (de 2ème traitement) travaillant en parallèle, c'est-à-dire le nombre des chaînes m₁ est fixé et m₁ > 1.

Dans ce cas :

- l'ordre des passages des types pour chaque chaîne de 2ème traitement est l'ordre décroissant de temps d'exécution de leurs objets (algorithme 4.1) pour l'ordonnancement de 2ème traitement),

- d'autre part les résultats du théorème (4.1) restent valables ainsi il suffit de remarquer :

$$\cdot \text{ si } (X_i)_{1 \leq i \leq m_1} = (x_{k1}^i)_{1 \leq k \leq p} \text{ la matrice représentant l'enga-}$$

gement journalier de chaque chaîne de 2ème traitement.

$$\text{Soit } X = (x_{k1})_{1 \leq k \leq p} = X_1 + X_2 + \dots + X_{m_1}$$

$$\text{avec } x_{k1} = \sum_{i=1}^{m_1} x_{k1}^i$$

$$\cdot \text{ d'autre part, pour chaque } i, 1 \leq i \leq m_1$$

$$\text{nous avons : } x_{11}^i \leq x_{21}^i \leq \dots \leq x_{k1}^i \leq x_{k+1,1}^i \leq \dots \leq x_{p1}^i$$

$$\text{d'où } \sum_{i=1}^{m_1} x_{11}^i \leq \sum_{i=1}^{m_1} x_{21}^i \leq \dots \leq \sum_{i=1}^{m_1} x_{k1}^i \leq \sum_{i=1}^{m_1} x_{k+1,1}^i \leq \dots \leq \sum_{i=1}^{m_1} x_{p1}^i$$

Il suffit maintenant de poser $Q_k = \sum_{i=1}^{m_1} x_{k1}^i \cdot b_1$ pour vérifier que le décalage est encore égal à un jour entre les deux traitements.

$$\text{Rappelons également que } S = |p_1| \cdot b_1 = \sum_{k=1}^p Q_k : p \text{ et}$$

$$Q_1 \leq Q_2 \leq \dots \leq Q_p$$

C/ Exemple numérique -

Nous avons choisi l'exemple de deux machines de Jonshon.

Rappel - Méthode de Johnson [3].

Problème : Un ensemble de pièces doit être produit sur deux machines, chaque machine peut traiter seulement une pièce à chaque instant et chaque pièce doit être traitée à travers la machine 1 puis à travers la machine 2.

Les gammes sont identiques et le critère à minimiser est le maximum des temps de passage. Johnson a montré que la séquence optimale est construite d'après la règle : "la pièce i précède la pièce j si

$$\underline{\text{Min}(A_i, B_j) < \text{Min}(A_j, B_i) "}$$

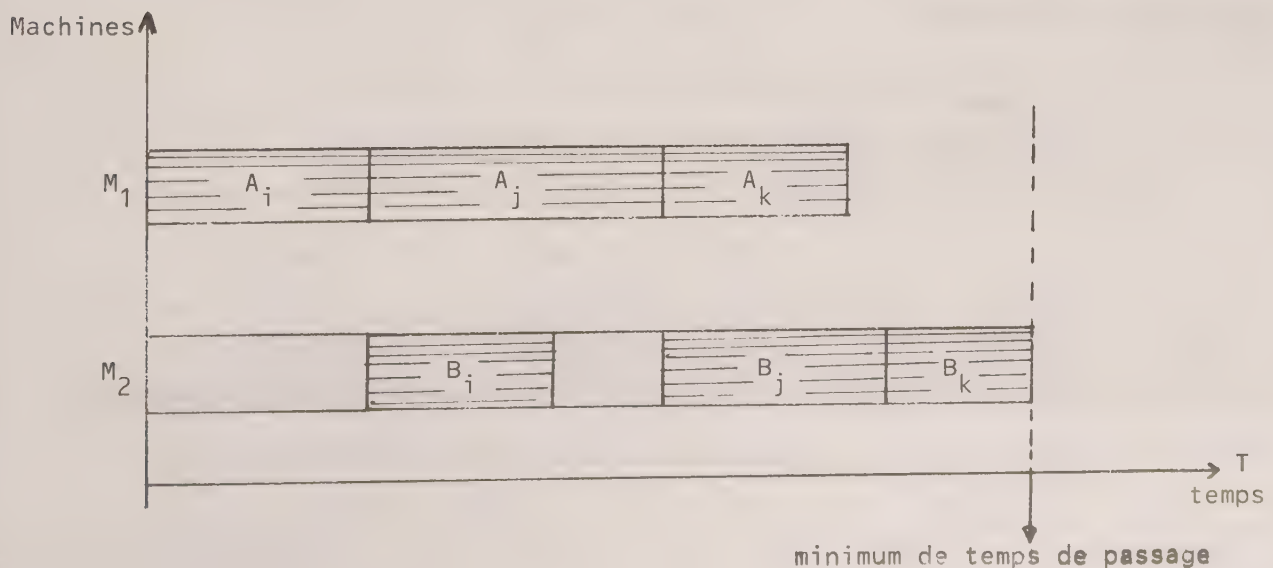
La transitivité est assurée en cas d'inégalité stricte, en cas d'égalité de $\text{min}(A_i, B_j)$ et $\text{min}(A_j, B_i)$ on prend soit i avant j, soit j avant i pour respecter la transitivité.

Avec A_i = le temps de traitement pour la ième pièce sur la 1ère machine,

B_i = le temps de traitement pour la ième pièce sur la 2ème machine.

Rappelons également que les ordres de passage des pièces sur les deux machines sont identiques.

Représentons graphiquement une solution qui satisfait à ces deux conditions :

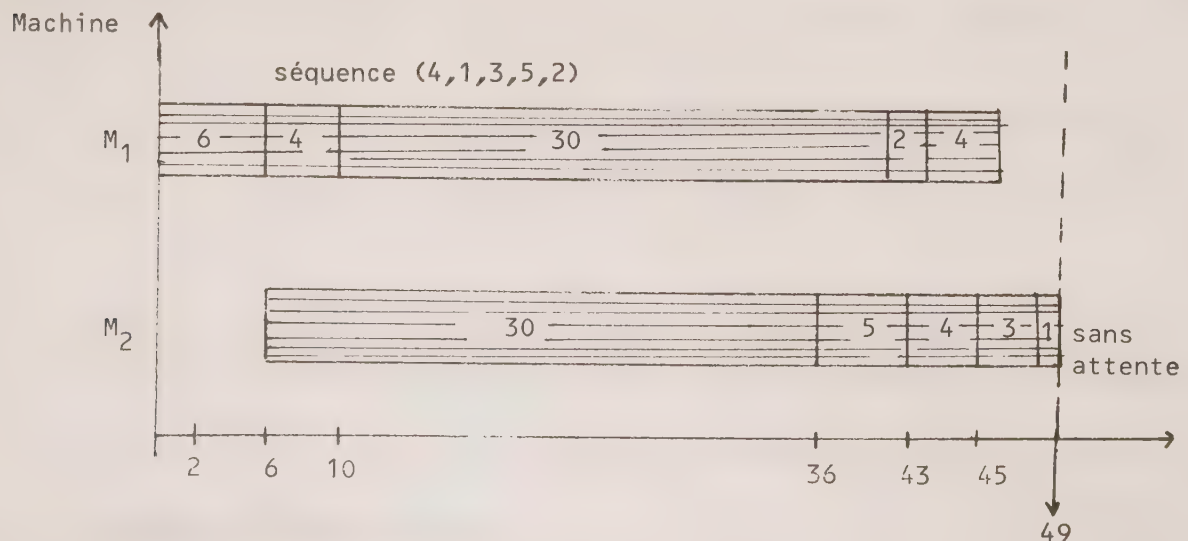


Pour illustrer sa méthode Johnson a donné l'exemple suivant :

i	A_i	B_i
1	4	5
2	4	1
3	30	4
4	6	30
5	2	3

Admettons maintenant qu'on veuille qu'il n'y ait pas d'attente dans le second traitement, il s'agit alors de trouver le décalage minimum entre les deux traitements. Notre algorithme 4.1. donne donc une solution ; ainsi dans l'exemple ci-dessus, dès que le traitement d'un objet est terminé sur la 1ère machine, on peut l'engager sur la 2ème machine. On considère que la 2ème machine de Johnson traite des types et la 1ère des parts. Etant donné que le temps de transition pour la 1ère machine est nul, on peut considérer que nous sommes dans le cas d'une part et m types. Alors l'algorithme 4.1. donne la séquence optimale qui est celle de l'ordre décroissant des $d_j = B_j$; c'est-à-dire la séquence (4,1,3,5,2). On remarque que cette solution n'est pas la même que celle de Johnson : (5,1,4,3,2). Le délai total pour cette séquence est de 4 unités et l'écoulement total de temps de 47 unités.

Représentation graphique -



Dans ce cas on peut vérifier le calcul numérique en adaptant la relation de dépendance entre les deux traitements (système général (1.3.2)) de la manière suivante :

On va considérer $S = 1$ unité de temps, alors $p=43$ unités de temps
 $p=43$ jours (2ème traitement)

$k =$ le jour de 2ème traitement (k ème unité

de temps à laquelle la pièce i est engagée dans la 2ème machine (2ème traitement). L'origine du temps est le début du travail de la 2ème machine. Le système s'écrit en appliquant le résultat de l'algorithme 4.1 avec $d =$ le nombre d'unités de travail dans le 1er traitement pour le 1er objet engagé dans le 2ème traitement (ici $d = 6$).

$$\forall k \in \{1, 2, \dots, 42\} = \{1, \dots, p-1\}$$

$$\sum_{l=1}^k Q_l \cdot b_l \leq (d+k) \times S \quad \text{la contrainte (OM2) n'existe pas.}$$

Pour le 1er objet engagé dans la 2ème machine, $k=1$

$$6 \leq (6+1) \times 1$$

Pour le 2ème objet engagé dans la 2ème machine, $k=30$

$$10 = 6+4 \leq (6+30) \times 1 = 36$$

Pour le 3ème objet engagé, $k=35$

$$40 = 6+4+30 \leq (6+35) \times 1 = 41$$

Pour le 4ème objet, $k=39$

$$42 = 6+4+30+2 \leq (6+39) \times 1 = 45$$

Pour le 5ème objet, $k=42$

$$46 = 6+4+30+2+4 \leq (6+42) \times 1 = 48$$

Le résultat reste valable si on exige la contrainte (OM2), c'est-à-dire

$$\sum_{l=1}^k Q_l \cdot b_l \leq (d+k-1) \times S.$$

On remarque que le temps de passage est 49 unités avec 0 unité de délai pour la 2ème machine, en effet dans notre problème une machine ne peut pas admettre une seule unité d'attente. Ceci était une 1ère différence entre les deux problèmes. D'autre part, dans le cas des algorithmes de flow-shop, on ne tient pas compte du temps de transition.

Exemple -

Soit trois pièces à traiter sur deux machines avec les données suivantes :

i	A_i	B_i
1	1	1
2	1	2
3	1	3

Si on applique l'algorithme de Johnson on obtient que toutes les séquences sont optimales puisque $\min(A_i, B_j) = \min(A_j, B_i) = 1$; ce qui est vrai. Mais ce résultat est faux si on complète le tableau précédent. De cette manière, en considérant toujours le cas d'une seule part et m types :

i	A_i	B_i	d_i	b_i
1	1	1	1	1
2	1	2	2	1
3	1	3	3	1

Le temps de passage d'un objet d'un type T_j à un objet d'un type

$$T_k \text{ est } e_{jk} = |d_j - d_k|.$$

Nous avons trois types différents, chaque type composé d'une seule pièce, de plus il y a une seule part ce qui explique l'égalité du temps d'exécution de chaque objet sur la 1ère machine.

Avec les données ci-dessus, il y a une seule séquence qui donne à la fois le nombre de jours de décalage minimum et le temps de transition minimum de 2ème traitement ; la séquence optimale est celle de l'algorithme 4.1. : (3, 2, 1).

II - SOLUTION DU PROBLEME $n|1|1,m_1|D1,(D2,T2.1)|C1 \text{ et } C2$ -

A/ Solution du problème $n|1|1,1|D1,D2|C1 \text{ et } C2$ -

(n part, 1 type, une seule chaîne par traitement, contraintes habituelles, minimiser temps de transition et nombre de jours de décalage).

On considère de plus que le temps de transition, si on passe d'un objet de la part p_i à un objet de la part p_k est $c_{i_k} = |b_i - b_k|$ où b_i est le temps d'exécution d'un objet de la part p_i . Comme le nombre des types est 1 donc le temps de transition de 2ème traitement est nul. Il reste à obtenir le temps de transition minimum pour le 1er traitement et le nombre de jours de décalage minimum d'où l'idée de l'algorithme suivant.

Algorithme 4.2. -

Soit $\mathcal{P} = \{p_1, p_2, \dots, p_h\}$ ensemble des parts, soit p_i et p_k deux éléments de $\mathcal{P}$ tel que $i \neq k$.

Alors la part p_k précède la part p_i dans le 1er traitement, en respectant le principe (1.1) travail par parts (respectivement dans le 2ème traitement).

Si $b_k < b_i$

(en cas d'égalité des b_i on pourra prendre l'ordre croissant ou décroissant des quantités d'objets dans les parts).

Théorème 4.2. -

L'algorithme 4.2. - réalise le temps de transition minimum,
- d'autre part l'ordonnancement est réalisable
pour un jour de décalage $d = 1$.

Démonstration -

Le temps de transition minimum pour le 1er traitement est atteint, si $b_1 \leq b_2 \leq \dots \leq b_n$ alors $\sum_{i=1}^{n-1} c_{i,i+1} = \sum_{i=1}^{n-1} |b_i - b_{i+1}| = b_n - b_1$.

Avec les notations du chapitre III nous avons :

$y_{11} \geq y_{21} \geq \dots \geq y_{k1} \geq y_{k+1} \geq \dots \geq y_{p1}$ (les parts prises dans l'ordre croissant des b_i)

où y_{k1} = le nombre d'objets en différentes parts engagés le k ème jour de 1er traitement dans le seul type (T_1)

L'engagement journalier moyen de 2ème traitement est :

$$S' = |T_1| \cdot d_1 : p = \sum_{k=1}^p y_{k1} \cdot d_1 : p \quad (\alpha)$$

On pose $Q_k = y_{k1} \cdot d_1$ d'où $Q_1 \geq Q_2 \geq \dots \geq Q_p$ (β)

On montre que le système (3.2) du chapitre III est vérifié pour $d=1$, c'est-à-dire :

$$\forall k \in \{1, \dots, p-1\} \quad \sum_{l=1}^k Q_l \geq (k-d+1) \cdot S' \quad (3.2)$$

avec $d=1$,

$$\forall k \in \{1, \dots, p-1\} \quad \sum_{l=1}^k Q_l \geq k \cdot S'$$

d'abord $Q_1 > S'$ sinon (α) serait faux.

Soit k fixé, $k \in \{1, \dots, p\}$.

$$1/ \text{ Si } Q_k \geq S' \text{ alors } \sum_{l=1}^k Q_l \geq k \cdot S' \quad \text{d'après (β)}$$

$$2/ \text{ Si } Q_k < S' \text{ supposons que } \sum_{l=1}^k Q_l < k \cdot S' \quad \dots \quad (1)$$

$$\text{comme } S' > Q_k \geq Q_{k+1} \geq \dots \geq Q_p \text{ alors } \sum_{l=k+1}^p Q_l < (p-k) \cdot S' \quad \dots \quad (2)$$

(1) + (2) entraînent :

$$\sum_{l=1}^k Q_l + \sum_{l=k+1}^p Q_l < k \cdot S' + (p-k) \cdot S' \quad \text{d'où}$$

$$\sum_{l=1}^p Q_l < p \cdot S' \quad \text{absurde d'après (α)}. \quad \text{Donc } \sum_{l=1}^k Q_l \geq k \cdot S' \text{ et on a bien}$$

le nombre de jours de décalage minimum $d = 1$.

B/ Exemple numérique -

Nous allons appliquer ce résultat à l'exemple précédent de deux machines de Johnson [3].

L'ordre croissant des $b_i = A_i$ donne la séquence :

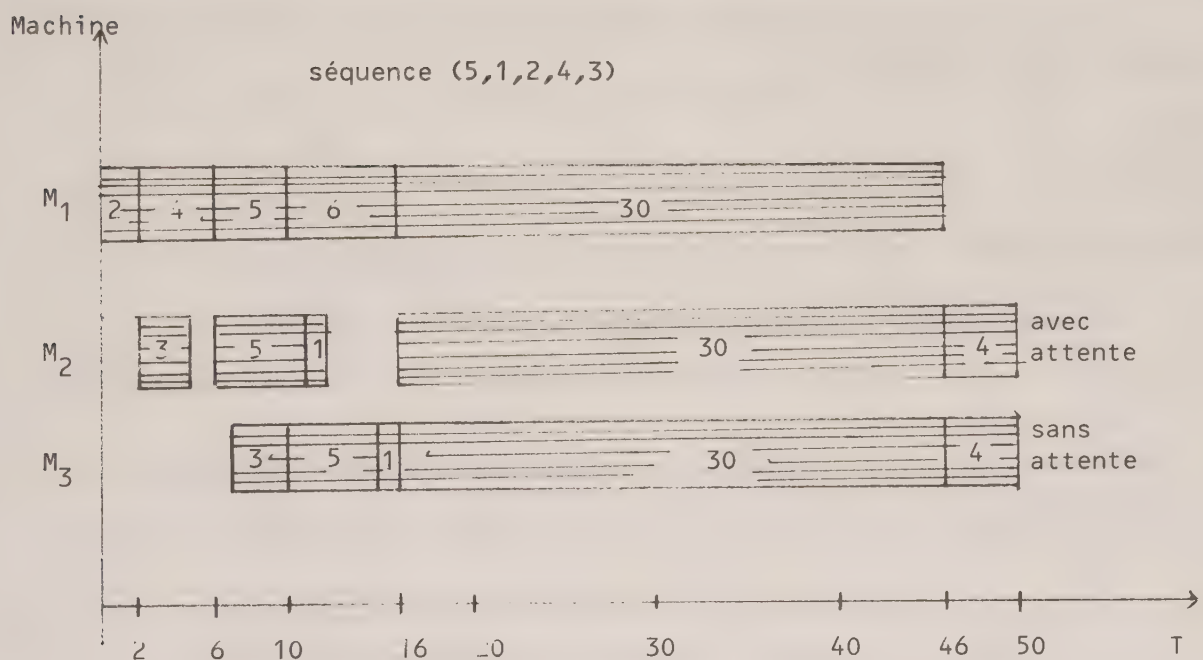
5, 1 ou 2, 4, 3 ; pour choisir si la pièce 1 précède 2 ou si la pièce 2 précède la pièce 1, nous avons deux méthodes.

Méthode 1 - Prendre la séquence donnant le nombre d'unités le plus petit comme décalage en appliquant (3.2) une fois avec 5, 1, 2, 4, 3 ; puis une autre avec 5, 2, 1, 4, 3.

Méthode 2 - En cas d'égalité de temps de traitement dans la 1ère chaîne, alors la pièce j précède la pièce k si $B_j > B_k$, si $B_j = B_k$ appliquer le système (3.2).

Nous allons appliquer la méthode 2 malgré que la 1ère, du point de vue mathématique est la plus juste. Donc nous avons la séquence (5, 1, 2, 4, 3).

Représentation graphique -



C/ Solution du problème $n|1|1,m_1|D1,(D2.T2.1)|C1$ et $C2$ -

Il y a m_1 chaînes de 2ème traitement, travaillant en parallèle. Le nombre de types par chaîne est 1. Ce problème est déjà résolu dans les : chapitre II :

En effet, dans ce chapitre on considère le temps de transition dans le 1er traitement ne dépendant pas des parts. C'est une constante c . chapitre III :

Si on considère le temps de transition (dans le 1er traitement) dépendant des parts, la matrice de transition est symétrique.

Ce problème avec m_1 chaînes, un type, n parts, est le même problème : une chaîne par traitement, n parts, m_1 types, avec les notations habituelles :

$$n|1|1,m_1|D1,(D2,T2.1)|C1 \text{ et } C2 \approx n|m_1|1,1|D1,(D2,T2)|C1.1 \text{ et } C2$$

III - PROBLEME $n \times m$ FLOW-SHOP MACHINE -

A/ Position du problème -

Le nombre de tâches est n ; le nombre de machines est m .
Les tâches (pièces) sont traitées dans les machines $M_1, M_2, \dots, M_m$ dans cet ordre, les ordres de passage des pièces sur les m machines sont identiques.

Soit la matrice suivante représentant les m machines et les n pièces avec leur temps d'exécution.

machine	M_1	M_2	M_3				M_m
pièces							
1	t_{11}	t_{12}	t_{13}				t_{1m}
2	t_{21}	t_{22}	t_{23}				t_{2m}
k	t_{k1}	t_{k2}	t_{k3}				t_{km}
n	t_{n1}	t_{n2}	t_{n3}				t_{nm}

t_{kj} = le temps d'exécution de la pièce k sur la machine j .

D'après le paragraphe I et II nous avons constaté dans le cas de deux machines que pour trouver une séquence admettant un décalage "minimum" en unité de temps, il fallait choisir entre deux séquences, une selon l'ordre croissant de temps de traitement dans la 1ère machine ;

L'autre selon l'ordre décroissant de temps de traitement dans la 2ème machine. Cette constatation nous a encouragé à proposer une heuristique qu'il faudrait peut être développer. Si on regarde le tableau ci-dessus on en déduit les remarques suivantes :

1/ pour que la $(j+1)$ ème machine ait un décalage le plus petit
 $1 \leq j \leq n-1$

par rapport à la j ème machine, (au début du traitement), il faut que la 1ère pièce engagée dans la j ème machine ait un temps de traitement égal à :

$$\min_{k=1}^n \{t_{kj}\} .$$

2/ pour que la $(j+1)$ ème machine admette un décalage le plus petit
 $1 \leq j \leq n-1$

par rapport à la 1ère machine (au début du traitement) il faut que la 1ère pièce engagée dans la 1ère machine ait un temps de traitement égal à :

$$\min_{k=1}^n \{ \sum_{i=1}^j t_{ki} \} .$$

3/ pour que la j ème machine ait un nombre d'unités de temps d'atten-
 $2 \leq j \leq n$

te minimum, il faudrait engager les n pièces dans cette machine selon l'ordre décroissant de leur temps de traitement.

Les trois remarques précédentes nous conduisent à la conclusion intuitive suivante : une "bonne séquence" est de prendre les objets selon l'ordre :

- croissant selon la remarque 1 et la remarque 2,
 - décroissant selon la remarque 3,
 - ou bien une succession d'ordre croissant, décroissant, croissant,...
- selon les remarques 1, 2 et 3.

Nous allons déterminer trois "bonnes séquences" et nous choisirons celle admettant le plus petit décalage. On s'est contenté de tenir compte des remarques précédentes que pour la 1ère machine, la m ème machine et la $(m-1)$ ème machine, d'où l'heuristique suivante :

B/ Heuristique -

Algorithme 4.3. -

Etape 1. -

- soit la séquence σ^1 (resp. σ^2) qui consiste à prendre les pièces dans l'ordre croissant des $t_{k,1}$ (resp. $\sum_{j=1}^{m-1} t_{k,j}$).

$$1 \leq k \leq n \quad 1 \leq k \leq n$$

soit $\sigma^1 = (x_1, 0, x_2, 0, \dots, x_{n-1}, 0, x_n, 0)$ $x_i \in \{1, \dots, n\}$
 $1 \leq i \leq n$

soit $\sigma^2 = (0, y_1, 0, y_2, \dots, y_{n-1}, 0, y_n)$ $y_i \in \{1, \dots, n\}$
 $1 \leq i \leq n$

soit $\sigma^1 \sigma^2 = \sigma^1 + \sigma^2 = (x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_k, y_k, \dots, x_n, y_n)$

soit $\sigma = (z_1, z_2, \dots, z_n)$ c'est les n premiers éléments différents (pris dans l'ordre) de la séquence $\sigma^1 \sigma^2$.

Etape 2. -

soit la séquence τ^1 (resp. τ^2) qui consiste à prendre les pièces dans l'ordre décroissant des $t_{k,m-1}$ (resp. $t_{k,m}$)

$$1 \leq k \leq n$$

Soit $\tau^1 = (x'_1, 0, x'_2, 0, \dots, x'_n, 0)$

soit $\tau^2 = (0, y'_1, 0, y'_2, \dots, 0, y'_n)$

soit $\tau^1 \tau^2 = \tau^1 + \tau^2 = (x'_1, y'_1, x'_2, y'_2, \dots, x'_k, y'_k, \dots, x'_n, y'_n)$

soit $\tau = (z'_1, z'_2, \dots, z'_n)$ c'est les n premiers éléments différents (pris dans l'ordre) de la séquence $\tau^1 \tau^2$.

Etape 3. -

soit $L = \left\lceil \frac{n}{m} \right\rceil$, soit la séquence θ déterminée ainsi avec

$$\theta = (u_1, u_2, \dots, u_L, u_{L+1}, u_{L+2}, \dots, u_{2L}, u_{2L+1}, \dots, u_n) \text{ avec } (2L+1 \leq n)$$

pour $i=1, \dots, L$; $u_i = x_i$;

pour $i=L+1, \dots, 2L$; $u_i = y_i'$ (non déjà pris)

pour $i=2L+1, \dots, n$; $u_i = x'_{n-(k-1)}$ pour $i=2L+k$ (non déjà pris)
 $1 \leq k \leq n-2L$

Etape 4. -

Calculer le décalage pour les séquences σ , τ , θ et prendre celle admettant le décalage le plus petit.

Calcul de décalage -

Le décalage de trois séquences σ , τ , θ est déterminé ainsi :

soit $j \in \{1, \dots, m\}$ j ième machine,

soit $\rho \in \{\sigma, \tau, \theta\}$

soit $\rho = (\rho_1, \rho_2, \dots, \rho_k, \rho_{k+1}, \dots, \rho_n)$ $\rho_k \in \{1, \dots, n\}$

On note par $\sum_{A^k} t_{\rho_k, j} = t_{\rho_1, j} + t_{\rho_2, j} + \dots + t_{\rho_k, j}$, $\sum_{\emptyset} t_{\rho_k, j} = 0$

On note par $d_k^j = \sum_{A^k} t_{\rho_k, j} - \sum_{A^{k-1}} t_{\rho_{k-1}, j+1}$
 $1 \leq j \leq n-1$

alors le décalage de la j ième machine pour que la $(j+1)$ ième machine ait 0 unité d'attente est :

$$d^j = \max_{k=1}^n \{d_k^j\}$$

On définit le décalage pour la séquence ρ ainsi :

$$d_\rho = \sum_{j=1}^{m-1} d^j$$

Soit $\rho_0 \in \{\sigma, \tau, \theta\}$ telle que $d_{\rho_0} = \inf_{\rho \in \{\sigma, \tau, \theta\}} \{d_\rho\}$

la séquence ρ_0 est la séquence déterminée par l'algorithme 4.3. Nous allons appliquer l'algorithme 4.3 sur des exemples pris aux articles traitant le

problème $n \times m$ flow-shop machines, où $m = 3, 4$ et 5 machines.

C/ Application -

Exemple 1 -

C'est l'exemple donné par JANTINDER N.D. GUPTA [16]

Où $n = 4$ pièces, et $m = 5$ machines

$i \backslash M$	M_1	M_2	M_3	M_4	M_5
1	4	3	7	2	8
2	3	7	2	8	5
3	1	2	4	3	7
4	3	4	3	7	2

Calcul numérique -

Etape 1. -

La séquence σ^1 selon l'ordre croissant de la 1ère machine :

$$\sigma^1 = (3, 0, 2, 0, 4, 0, 1, 0) \text{ ou } \sigma^1 = (3, 0, 4, 0, 2, 0, 1, 0) \text{ car } t_{2,1} = t_{4,1} = 3$$

La séquence σ^2 selon l'ordre croissant des $\sum_{j=1}^{m-1} t_{k,j}$

$$\text{pour la pièce 1 nous avons } \sum_{j=1}^4 t_{1,j} = 16$$

$$\text{pour la pièce 2 nous avons } \sum_{j=1}^4 t_{2,j} = 20$$

$$\text{pour la pièce 3 nous avons } \sum_{j=1}^4 t_{3,j} = 10$$

$$\text{pour la pièce 4 nous avons } \sum_{j=1}^4 t_{4,j} = 17$$

$$\text{d'où : } \sigma^2 = (0, 3, 0, 1, 0, 4, 0, 2)$$

donc : $\sigma^1 \sigma_2 = (3,3,2,1,4,4,1,2)$ donc $\sigma = (3,2,1,4)$

$\sigma_1^1 \sigma_2 = (3,3,4,1,2,4,1,2)$ donc $\sigma^1 = (3,4,1,2)$

Etape 2. -

La séquence τ^1 selon l'ordre décroissant de la (m-1)ième machine :

$$\tau^1 = (2,0,4,0,3,0,1,0)$$

La séquence τ^2 selon l'ordre décroissant de la dernière machine :

$$\tau^2 = (0,1,0,3,0,2,0,4)$$

d'où $\tau^1 \tau^2 = (2,1,4,3,3,2,1,4)$ donc $\tau = (2,1,4,3)$

Etape 3. -

$$L = \left\lceil \frac{n}{m} \right\rceil = \left\lceil \frac{4}{5} \right\rceil = 1$$

d'où $\theta = (3,1,4,2)$

Etape 4. -

Calcul de décalage.

1/ Séquence $\sigma = (3,2,4,1)$

- décalage entre la 1ère chaîne et la 2ème chaîne -

$$d_1^1 = t_{3,1} - 0 = 1 ;$$

$$d_2^1 = t_{3,1} + t_{2,1} - t_{3,2} = 1+3-2 = 2$$

$$\begin{aligned} d_3^1 &= \sum_{A^k} t_{k,1} - \sum_{A^{k-1}} t_{k,2} = t_{3,1} + t_{2,1} + t_{4,1} - (t_{3,2} + t_{2,2}) \\ &= 1+3+3-(2+7) = 7 - 9 = -2 ; \end{aligned}$$

$$\begin{aligned} d_4^1 &= t_{3,1} + t_{2,1} + t_{4,1} + t_{1,1} - (t_{3,2} + t_{2,2} + t_{4,2}) = \\ &= 1+3+3+4 - (2+7+4) = 11 - 13 = -2 ; \end{aligned}$$

$$\text{d'où } d^1 = \text{Max } \{d_1^1, d_2^1, d_3^1, d_4^1\} = \text{Max } \{1, 2, -2, -2\} = 2.$$

- décalage entre la 2ème chaîne et la 3ème chaîne, toujours avec $\sigma = (3, 2, 4, 1)$:

$$d_1^2 = t_{3,2}^{-0} = 2$$

$$d_2^2 = t_{3,2} + t_{2,2} - t_{3,3} = 2+7-4 = 9-4 = 5$$

$$d_3^2 = t_{3,2} + t_{2,2} + t_{4,2} - (t_{3,3} + t_{2,3}) = 2+7+4 - (4+2) = 7$$

$$\begin{aligned} d_4^2 &= t_{3,2} + t_{2,2} + t_{4,2} + t_{1,2} - (t_{3,3} + t_{2,3} + t_{4,3}) = 2+7+4+3 - (4+2+3) \\ &= 16-9 = 7 \end{aligned}$$

$$\text{donc } d^2 = \text{Max } (2, 5, 7, 7) = 7$$

- décalage entre la 3ème chaîne et la 4ème chaîne toujours avec $\sigma = (3, 2, 4, 1)$:

$$d_1^3 = t_{3,3}^{-0} = 4 ; \quad d_2^3 = t_{3,3} + t_{2,3} - t_{3,9} = 4+2-3 = 3 ;$$

$$d_3^3 = t_{3,3} + t_{2,3} + t_{4,3} - (t_{3,4} + t_{2,4}) = 4+2+3 - (3+8) = -2 ;$$

$$d_4^3 = t_{3,3} + t_{2,3} + t_{4,3} + t_{1,3} - (t_{3,4} + t_{2,4} + t_{4,3}) =$$

$$(4+2+3+7) - (3+8+3) = 2$$

$$\text{d'où } d^3 = \text{Max } (4, 3, -2, 2) = 4$$

- décalage entre la 4ème machine et la 5ème machine :

$$d_1^4 = t_{3,4} - 0 = 3$$

$$d_2^4 = t_{3,4} + t_{2,4} - t_{3,5} = 3+8-7 = 4$$

$$d_3^4 = t_{3,4} + t_{2,4} + t_{4,4} - (t_{3,5} + t_{4,5}) = 3+8+7 - (7+5) = 6$$

$$\text{d'où } d^4 = \text{Max } (3, 4, 6, 6) = 6$$

Alors le décalage pour σ est : $d^1 + d^2 + d^3 + d^4 = 2+7+4+6 = 19$;

2/ De même manière on calcule le décalage pour la séquence $\sigma^1 = (3, 4, 1, 2)$ -

$$\text{On trouve } d^1 = \text{Max } \{1, 2, 2, 2\} = 2$$

$$d^2 = \text{Max } \{2, 2, 2, 2\} = 2$$

$$d^3 = \{4, 4, 4, 4\} = 4$$

$$d^4 = \{3, 3, 3, 3\} = 3$$

Donc le décalage pour σ^1 est $d^1 + d^2 + d^3 + d^4 = 11$

donc la séquence choisie est σ^1 de décalage 11, $\sigma^1 = (3, 4, 1, 2)$.

3/ Comme précédemment on calcule le décalage pour la séquence $\tau = (2, 1, 4, 3)$. Le décalage entre les machines est : $d^1 = 3$, $d^2 = 9$, $d^3 = 2$, $d^4 = 8$.

D'où le décalage pour la séquence τ est :

$$d^1 + d^2 + d^3 + d_4 = 3 + 9 + 2 + 8 = 21.$$

4/ Décalage pour la séquence $\theta = (3, 1, 4, 2)$

Décalage entre les machines : $d^1=3$, $d^2=2$, $d^3=9$, $d^4=3$

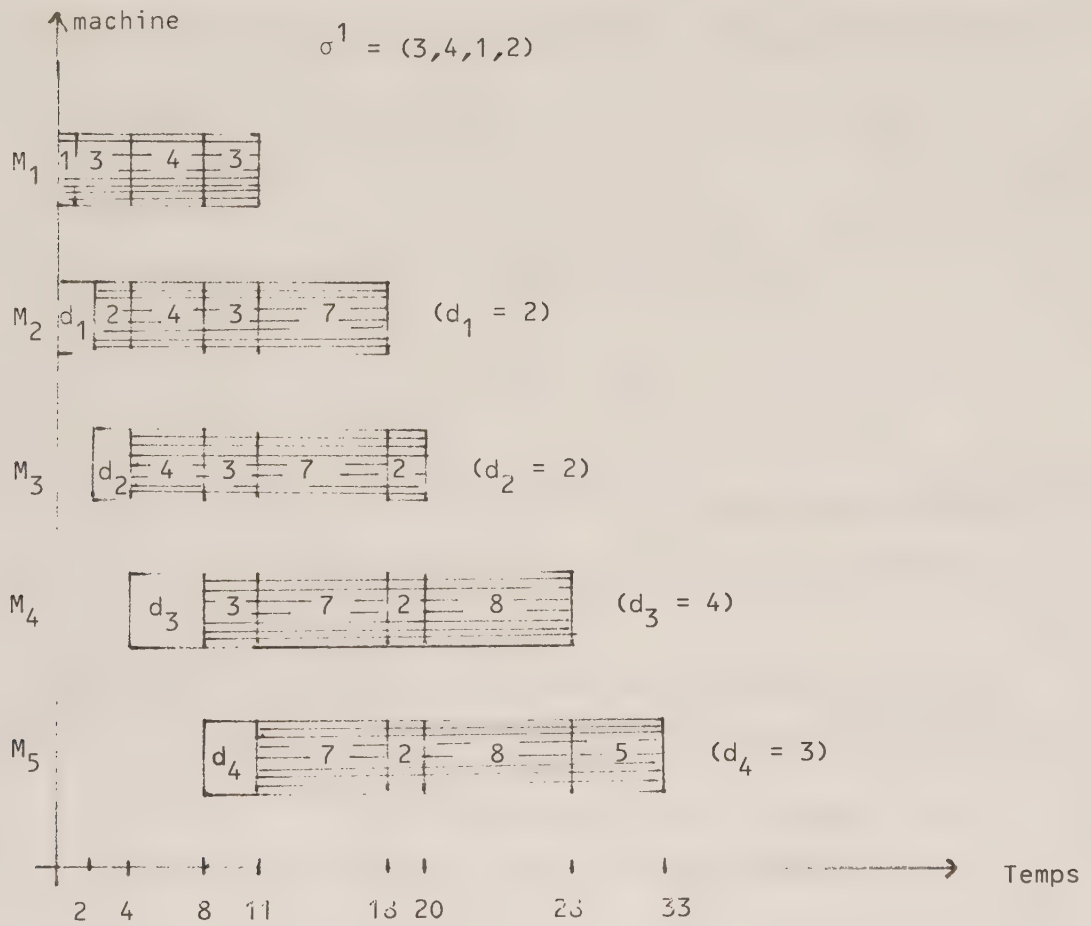
d'où le décalage pour θ est : $3+2+9+3 = 17$ unités.

Conclusion -

La séquence optimale est $\sigma^1 = (3, 4, 1, 2)$ de décalage 11 unités.

σ^1 est la séquence donnée par N.D.GUPTA de temps total de passage égal à 33 unités.

Représentation graphique de σ^1 sans attente -



Exemple 2. -

C'est l'exemple donné par M.L. SMITH, S.S. PANWALKAR and R.A. DUDEK [17].

où $n = 4$ pièces et $m = 4$ machines.

$i \backslash M$	M_1	M_2	M_3	M_4
1	23	16	20	25
2	18	12	15	21
3	25	22	24	30
4	13	8	12	18

Calcul numérique -Etape 1. -

$$\sigma^1 = (4, 0, 2, 0, 1, 0, 3, 0) \text{ ordre croissant de } t_{k,1} \quad (k=1,2,3,4)$$

$$\sigma^2 = (0, 4, 0, 2, 0, 1, 0, 3) \text{ ordre croissant de } \sum_{j=1}^3 t_{k,j} \quad (k=1,2,3,4)$$

d'où $\sigma = (4, 2, 1, 3)$ c'est la séquence déjà trouvée dans [17].

Etape 2. -

$$\tau^1 = (3, 0, 1, 0, 2, 0, 4, 0) \text{ ordre décroissant de } t_{k,3} \quad (k=1,2,3,4)$$

$$\tau^2 = (0, 3, 0, 1, 0, 2, 0, 4) \text{ ordre décroissant de } t_{k,4} \quad (k=1,2,3,4)$$

d'où $\tau = (3, 1, 2, 4)$

Etape 3. -

$$L = \left\lceil \frac{n}{m} \right\rceil = \left\lceil \frac{4}{4} \right\rceil = 1 \text{ d'où}$$

$$\theta = (4, 3, 2, 1)$$

Etape 4. -

Calcule de décalage pour σ , τ et θ .

décalage pour σ :

$$\text{décalage entre les machines } d^1 = \text{Max} \{13, 23, 34, 43\} = 43 ;$$

$$d^2 = \text{Max} \{8, 8, 9, 11\} = 11 ;$$

$$d^3 = \text{Max} \{12, 9, 8, 7\} = 12$$

$$\text{Donc le décalage pour } \sigma \text{ est } d^1 + d^2 + d^3 = 43 + 11 + 12 = 66$$

C'est la séquence optimale puisque le décalage pour τ est de 74 unités, et pour θ est supérieur à 100.

Exemple 3 (n jobs, 3 machines) -

C'est l'exemple donné par FENNELL BURNS et John ROOKER.

A special case of 3xn flow shop problem [18]; où $n = 8$ jobs et $m=3$ machines.

M \ i	1	2	3	4	5	6	7	8
M_1	1	7	7	4	1	6	17	17
M_2	10	7	6	5	9	6	15	15
M_3	12	3	3	5	11	2	16	17

La séquence optimale en appliquant l'algorithme 4.3. est la séquence θ de l'étape 3., c'est-à-dire en prenant les objets dans l'ordre croissant de M_1 , décroissant de M_3 , croissant de M_2 .

Nous avons $L = \left\lceil \frac{8}{3} \right\rceil = 3$, c'est-à-dire :

u_1, u_2, u_L est 1 ou 5 puis 4 (car $t_{1,1} = t_{5,1}$)

$u_{L+1}, u_{L+2}, u_{L+3}$ et u_{L+4} est 8,7,3 ou 2 (car $t_{3,3} = t_{2,3}$)

u_{2L+2} est 6, c'est le seul non encore pris.

La séquence admettant le décalage le plus petit est $\theta = (5,1,4,8,7,2 \text{ ou } 3, 6)$ avec un décalage égal à 12 unités de temps.

Dans [18] la séquence optimale est (5,1,4,8,7,2,3,6).

Remarque -

Nous avons fait des essais numériques avec d'autres exemples, citons les deux seuls exemples où la séquence optimale non trouvée [19,20], on trouve une séquence dans chaque cas admettant une seule unité de délai, supérieure au temps total de passage de la séquence optimale.

CHAPITRE VDECALAGE FIXE

Ce chapitre est destiné à déterminer l'ordonnancement de 1er traitement avec la nouvelle contrainte (T3) le nombre de jours de décalage entre les deux traitements est fixé. De plus la contrainte (T2) l'ordonnancement de 2ème traitement est fixé. On cherche la solution du problème $n|m|1,m_1|(D1, T3),(D2, T2, T2.1)|C1.1$; nous présentons deux bons heuristiques.

I/ MARCHE EN ARRIERE -

L'algorithme que nous détaillons ici est dû à M. TREHEL [2]

A) Visualisation des contraintes -

Soit p le nombre de jours de la période ; d est le nombre de jours de décalage. Si le 1er traitement commence à la date 1, le 2ème traitement commence à la date $d+1$;

Considérons que :

Le 1er jour de 2ème traitement, on engage la partie Z_1

Le 2ème jour de 2ème traitement, on engage la partie Z_2

.

.

.

Le kème jour de 2ème traitement, on engage la partie Z_k

.

.

.

Le pème jour de 2ème traitement, on engage la partie Z_p

Connaissant $Z_1, Z_2, \dots, Z_k, \dots, Z_p$, il s'agit de trouver les inconnus $U_1, U_2, \dots, U_p$, qui seront les parties à engager dans le 1er traitement, chacun de p jours de la période.

1er traitement			2ème traitement	
j	Stock avant le début du j ^{ème} jour	engagement du j ^{ème} jour	Stock avant le début du j ^{ème} jour	engagement du j ^{ème} jour
1	$U_1 U_2 \dots U_p$	U_1		
2	$U_2 \dots U_p$	U_2		
.	.	.		
.	.	.		
d	$U_d \dots U_p$	U_d	$U_1 U_2 \dots U_{d-1}$	
d+1	$U_{d+1} \dots U_p$	U_{d+1}	$U_1 U_2 \dots U_d$	Z_1
.	.	.	$U_1 U_2 \dots U_{d+1} - Z_1$	Z_2
.	.	.	.	.
.	.	.	.	.
k	$U_k \dots U_p$	U_k	$U_1 \dots U_{k-1} - Z_1 Z_2 \dots Z_{h-1}$	Z_h
.	.	.	.	.
.	.	.	.	.
p	U_p	U_p	$U_1 \dots U_{p-1} - Z_1 \dots Z_{p-d-1}$	Z_{p-d}
p+1			$U_1 \dots U_p - Z_1 \dots Z_{p-d}$	Z_{p-d+1}
.			.	.
.			.	.
.			.	.
p+d			$U_1 \dots U_p - Z_1 \dots Z_{p-1} = Z_p$	Z_p

Note : on note par $Z_1 \cdot Z_2 \dots Z_k = Z_1 U Z_2 U \dots U Z_k$

On constate que les contraintes à respecter sont :

- Tailles de $U_1, U_2, \dots, U_p$; (égales en temps de traitement), on notera Q cette taille qui est la capacité par jour.
- $U_1 U_2 \dots U_k$ contient $Z_1 Z_2 \dots Z_h$ ($k = h+d$)

Cette dernière contrainte se généralise ainsi :

les parts faites sur les $(d+h-1)$ premiers jours contiennent la partie à engager du h ème jour de 2ème traitement. Compte tenu de cette contrainte, nous avons à minimiser le nombre de changement de parts. (ici le temps de transition si on passe d'un objet de la part P_i à une part P_k dépend de parts, c'est une constante).

B) Principe de l'algorithme -

On commence par travailler sur l'ensemble des $(p-1)$ premiers jours de 1er traitement, on essaie de faire dans ces $(p-1)$ premiers jours toutes les parts π pour lesquelles au moins un objet à engager dans les $(p-d)$ premiers jours de 2ème traitement de part π .

Ceci signifie que pour chaque part impliquée dans l'engagement (de 2ème traitement) l'un des $(p-d)$ premiers jours ; on fait ses objets dans le 1er traitement sans interruption.

Si cet ensemble d'objets (à traiter dans le 1er traitement) est de taille inférieure ou égale à $(p-1)Q$, il n'y aura pas lors de p ème jour nécessité de revenir sur une part déjà faite l'un des $(p-1)$ premiers jours.

Si ce n'est pas le cas, il y aura au moins un changement. Un module doit alors déterminer les parts à traiter complètement et celles pour lesquelles il ne faut traiter que ce qui est utile aux $(p-d)$ premiers jours de 2ème traitement.

En complétant éventuellement à $(p-1)Q$, on définit ainsi l'ensemble des objets à traiter dans les $(p-1)$ premiers jours. On représente cet ensemble sous la forme d'une nouvelle matrice. On applique alors la même procédure à partir de cette matrice pour les

(p-2) premiers jours de la période, avec les contraintes des (p-d-1) jours de 2ème traitement. On détermine ainsi de proche en proche ce que nous avons appelé $U_p, U_{p-1}, \dots, U_2, U_1$.

C) Détail de l'algorithme -

Données initiales

- la matrice des besoins,
- l'ordonnancement en type de 2ème traitement.

Etapas

Chacune des étapes détermine :

- l'ensemble qui suit la période concernée,
- la matrice des besoins, qui sera une donnée pour l'étape suivante.

Avec le schéma ci-dessus:

- la première étape part de $U_1 U_2 \dots U_p$, détermine une partie

$U_1 U_2 \dots U_{p-1}$, avec la contrainte

$$Z_1 Z_2 \dots Z_{p-d} \subseteq U_1 U_2 \dots U_p,$$

Elle donne donc $Z_{p-d+1} \dots Z_p$ par différence.

- la deuxième étape part de $U_1 U_2 \dots U_{p-1}$, détermine une partie

$U_1 U_2, \dots, U_{p-2}$, avec la contrainte

$$Z_1 Z_2 \dots Z_{p-d-1} \subseteq U_1 U_2 \dots U_{p-1}$$

Elle donne donc Z_{p-d} par différence.

- d'une manière générale, la hième étape part de $U_1 U_2 \dots U_k$ détermine une partie $U_1 U_2, \dots, U_{k-1}$ avec la contrainte

$$Z_1 Z_2 \dots Z_{k-1} \subseteq U_1 U_2 \dots U_{k-1} \quad (k=h+d)$$

Elle donne donc Z_h par différence.

Généralement il y a p-d étapes.

Définition :

Un type est dit complet si tous ses objets sont traités le même jour de 2ème traitement, sinon le type est dit incomplet.

1) Détermination des parts pour les types complets et incomplets -

Pour déterminer les parts, nous adoptons la procédure suivante :

- détermination des parts des types complets pour la période intéressée;
- pour chaque type incomplet : prendre comme part les parts déterminées ci-dessus. Le nombre d'objets à prendre dans ces parts est dans la limite des besoins. S'il reste des objets à prendre dans d'autres parts, enregistrer pour un module à côté le nombre d'objets, les parts possibles, les besoins dans chaque part.

Il s'agit maintenant de choisir le nombre minimum de parts en plus des parts déterminées ci-dessus.

Nous n'avons pas pour le moment d'autre algorithme que celui qui consiste à énumérer tous les sous-ensembles des parts de l'ensemble des parts concernées. Pour chaque sous-ensemble, on examine si ce sous-ensemble suffit pour épuiser les types incomplets.

On énumérera dans l'ordre croissant des tailles des sous-ensembles et pour une taille donnée dans l'ordre lexicographique. On est certain d'avoir le minimum.

2) Nous avons précisé l'ordonnancement des chaînes de 2ème traitement en fonction des parts ; il s'agit maintenant de définir ce qu'il faut traiter dans les (p-h) premiers jours.

Notons $1, 2, \dots, s$ les parts concernées dans le p+1-d-h premiers jours de 2ème traitement, et $a_1, a_2, \dots, a_s$ les quantités d'objets à engager dans ces p+1-d-h jours.

$$\text{On a } \sum_{k=1}^s a_k \leq (p-h)Q$$

puisque $(p-h)Q$ correspond à la capacité de 1er traitement pour $(p-h)$ jours,

et $\sum_{k=1}^s a_k$ à la capacité de 2ème traitement pour $p+1-d-h$ jours.

Soit la capacité résiduelle de 1er traitement

$$C_r = (p-h)Q - \sum_{k=1}^s a_k$$

Soit pour $j=1,2,\dots,s$, b_j la différence entre le nombre total d'objets de la part p_j et a_j

Si toutes les parts $P_1, \dots, P_s$ peuvent être traitées dans les $(p-h)$ premiers jours, on n'aura pas le $(p-h+1)$ ième jours à reprendre une part prise pour traiter les $\sum_{k=1}^s a_k$ objets ; tel ne sera en général

pas le cas, il faut donc faute de terminer toutes les parts, en terminer le plus possible.

On peut écrire : Maximiser $\sum_{j=1}^s x_j$ avec $x_j = 0$ ou 1

$$\text{et } \sum_{j=1}^s x_j \cdot b_j \leq C_r$$

Ceci se résoud en prenant $x_j=1$ pour les b_j successifs dans l'ordre croissant tant que la somme ne dépasse pas C_r et $x_j=0$ sinon le nombre de parts à reprendre le $(p-h+1)$ ième jour sera

$$s - \sum_{j=1}^s x_j - 1$$

Le nombre d'objets à traiter pour l'ensemble des $p-h$ premiers jours est :

$$a_j + x_j \cdot b_j \quad \text{pour } j = 1, 2, \dots, s.$$

qu'on complète par une partie des b_j pour le j suivant (dans l'ordre croissant des b_j), pour arriver à C_r

Pour le $(p-h+1)$ ième jour, on traite :

- la suite pour cette quantité b_j dont il vient d'être question
- les b_j pour lesquels $x_j=0$
- le reste de la matrice en groupant naturellement par parts.

La matrice des besoins pour l'étape suivante est obtenue en enlevant de la matrice des besoins de l'étape h les objets à traiter le $(p-h+1)$ ième jour. On passe alors à l'étape suivante et ceci jusque $h=p-d$

3) La $(p-d)$ ième étape a déterminé une matrice des besoins qui correspond à l'ensemble à traiter dans les d premiers jours.

L'ordre dans cet ensemble est quelconque, il faut seulement prendre les parts d'un seul coup et finir par une part présente le $(d+1)$ ième jour pour ne pas avoir de changement de part du dième au $(d+1)$ ième jour.

D) Application numérique -

Regarder l'exemple concret paragraphe III.

II/ MARCHE EN AVANT -

Avant de décrire l'algorithme de marche en avant, nous allons déterminer la borne supérieure de nombre de changements.

On suppose que le nombre de jours de décalage de 1er traitement d est fixée ; de plus l'ordonnancement en types de 2ème traitement est fixé, on utilise les mêmes notations des chapitres I et II.

a) On suppose que d vérifie l'engagement journalier de 2ème traitement, c.a.d. $\forall k \in \{1, \dots, p-d\}$ $\sum_{l=1}^k q_l \leq (d+k-1) \times s$.

b) On suppose que $d < d'$ entier minimum déterminée par l'algorithme (2.3.2) car si $d \geq d'$ alors le nombre de changement en part est minimum à savoir $(n-1)$ changements avec (n =nombre de parts) puisque on peut travailler par parts pour le 1er traitement.

A) Décomposition de la matrice représentant l'ordonnancement de 2ème traitement -

Soit $L \in \mathbb{N}^*$ tel que :

$$\sum_{l=1}^L q_l \leq d \times s \quad \text{et} \quad \sum_{l=1}^{L+1} q_l > d \times s$$

$$L \in \mathbb{N}^* \quad \text{car} \quad q_1 \leq d \times s$$

- 1 - Soit N_L = le nombre de parts distinctes passant dans les L premiers jours de 2ème traitement.
- 2 - Soit N_d = le nombre de parts distinctes passant dans les d derniers jours de 2ème traitement.
- 3 - Pour k , tel que $L+1 \leq k \leq p-d$, on note par N_k = le nombre de parts passant dans le k ème jour de 2ème traitement.

Soit X la matrice représentant l'ordonnancement de 2ème traitement en fonction des parts.

	p ₁	p ₂	p ₃		p _n	
BLOC 1	x ₁₁	x ₁₂	x ₁₃		x _{1n}	1er jour de 2ème traitement
	x ₂₁	x ₂₂	x ₂₃		x _L	Lième jour de 2ème traitement
	x _{k1}	x _{k2}	x _{k3}		x _{kn}	kième jour de 2ème traitement
BLOC 2	x _{p-d,1}	x _{p-d,2}	x _{p-d,3}		x _{p-d,n}	(p-d)ième jour de 2ème traitement
BLOC 3	x _{p1}	x _{p2}	x _{p3}		x _{pn}	pième jour de 2ème traitement

D'après la décomposition de la matrice X en trois blocs, et grâce qu nombre de jours de décalage de 1er traitement, on constate :

4 - qu'on peut travailler par part les d premiers jours de 1er traitement.

En effet, on peut traiter $\sum_{l=1}^L Q_l$ objets en NL parts distinctes, donc nous avons (NL-1) changements.

- 5 - On peut travailler par part à la fin de la période de 1er traitement en terminant le travail par $\sum_{L=p-d+1}^p Q_L$ objets en Nd parts distinctes.

B) Borne supérieure du nombre de changements-

Comme l'engagement journalier de 2ème traitement est respecté, on peut imposer l'ordonnancement suivant (c'est le plus mauvais en temps de transition) pour le 1er traitement.

On propose l'ordre de passage de parts suivant :

(i) faire les objets $\sum_{j=1}^L Q_j$ en NL parts en respectant (4)

(ii) enchainé par les objets des parts :

$$N_{L+1}, N_{L+2}, \dots, N_k, \dots, N_{p-d},$$

$$L+1 \leq k \leq p-d$$

(iii) terminer le 1er traitement par les objets $(\sum_{j=p-d+1}^p Q_j)$ en Nd part en respectant (5)

d'où l'ordre de passage de parts est :

$$NL, N_{L+1}, \dots, N_k, \dots, N_{p-d}, Nd.$$

De plus, on impose que le passage d'un ensemble ci-dessus à son successeur se fait sans changement en part dans le 1er traitement ; s'ils ont au moins une part en commun, sauf dans le cas suivant :

$$\text{card}(\text{pred}(X) \cap X \cap \text{succ}(X)) = 1$$

avec

$$\text{card } X \geq 2 \quad \text{et} \quad X \in \{N_{L+1}, \dots, N_{p-d}\}$$

On note par : succ = successeur

pred = prédécesseur

alors le passage d'un ensemble à un autre sans changement n'est réalisé qu'une fois.

De ce qui précède on en déduit que le nombre de changements en part est :

$$N \in S = (NL - 1) + \sum_{k=L+1}^{p-d} (N_k - 1) + (Nd - 1) + K'$$

où K' est le nombre de passage d'un ensemble à son successeur avec changement en part.

Le temps de transition du 1er traitement est :

$$C \times (NL + Nd + \sum_{k=L+1}^{p-d} (N_k - 1) + K' - 2)$$

ou

C = le temps de transition si on passe d'un objet de part P_i à un objet de part P_j .

Remarque.

Le nombre de changement en part NCS peut-être atteint.

C) Algorithme Marche en avant.

Rappelons que la Méthode Marche en arrière dans I) détermine l'ordonnancement pour le 1er traitement en $(p-d)$ étapes, la 1ère étape détermine l'ensemble d'objets à traiter dans les $(p-1)$ premiers jours, puis l'ensemble d'objets à traiter dans les $(p-2)$ premiers jours, puis pour les $(p-3), \dots$, puis pour les d premiers jours.

L'Algorithme Marche en avant que nous allons décrire, détermine l'ordonnancement du 1er traitement avec un nombre d'étapes $\leq (p-d)$.

La 1ère étape détermine l'ensemble d'objet à traiter les d premiers jours.

La 2ème étape détermine l'ensemble d'objet à traiter dans le $(d+r)$ premiers jours, $1 \leq r$; r est le nombre de jours de 2ème traitement dont leurs engagements journaliers sont satisfaits. La matrice X est réduite à $r+1, \dots, p-d+1$ lignes.

La dernière étape est celle qui déterminera l'ensemble d'objets contenant tous les objets (non encore pris) de la $(p-d)$ ième jour de 2ème traitement.

On utilisera le processus donné par M. TREHEL [2] pour déterminer les parts des types complet et incomplet ; le processus est appliqué aux $(p-d)$ premiers jours de 2ème traitement.

On détermine ainsi la matrice $X = X_1 + X_2 + \dots + X_{m1}$ représentant l'ordonnancement des chaînes de 2ème traitement en fonction des parts.

La matrice X admet n colonnes représentant les parts $P_1, P_2, \dots, P_n$.

La matrice X admet $(p-d+1)$ lignes.

La $(p-d+1)$ ième ligne est définie :

$$\text{pour tout } i \in \{1, \dots, n\} \quad x_{p-d+1, i} = \begin{cases} |P_i| & \text{si } P_i \in N_{p-d} \\ |P_i| - \sum_{l=1}^{p-d} x_{li} & \text{sinon} \end{cases}$$

ou x_{li} = le nombre d'objet en différents types engagés le l'ième jour de 2ème traitement en part P_i .

$$N_{p-d} = \bigcup_{k=1}^{p-d} \left(\bigcup_{j=1}^{m_k} N_j^k \right) \quad \text{ou } N_j^k = \text{ensemble des parts apparaissant le } k\text{ième jour de traitement dans la } j\text{ième chaîne.}$$

L'Algorithme Marche en avant est basé sur 4 étapes principales ; d'une manière générale, chaque étape s'écrit pour un kième jour de 2ème traitement de la façon suivante :

ETAPE 1 :

Appliquer l'Algorithme 2.3.2 à la matrice X pour trouver le système d'ordre h, admettant le nombre de jours de décalage minimum d', et l'ordonnancement de temps de transition minimum associé au système d'ordre h.

Soit Nd' le 1er ensemble des parts dans l'ordonnancement associé au système d'ordre h. (Voir Chapitre II).

si $h = 0$ alors $A_k \subseteq Nd'$ sinon $Nd' = A_k$

ou A_k est l'ensemble des parts apparaissant le kième jour de 2ème traitement.

ETAPE 2 :

Soit $Z := dxS - Q_k$, $Q_k = \sum_{i \in A_k} x_{ki} \cdot b_i$

$Z \geq 0$ car d vérifie l'engagement journalier du 2ème traitement.

si $Z \neq 0$, il s'agit maintenant de déterminer l'ensemble des objets accompagnant ceux de Q_k ; et ceci en maximisant le nombre de parts entièrement traitées dans les d premiers jours de 1er traitement.

On note par $|P_i| = |P_i| - x_{ki}$; $i \in A_k$;

soit NNd' le plus grand sous ensemble de Nd' , tel que :

$$\sum_{i \in NNd'} |P_i| \cdot b_i \leq Z ;$$

si pour tout $i \in Nd'$, $|P_i| \cdot b_i > Z$ alors $NNd' = \emptyset$;

ETAPE 3 :

$Z := Z - \sum_{i \in NNd'} |P_i| \cdot b_i$; pour épuiser les Z unités de temps, on

complètera d'abord par le nombre maximum des $x_{k+1,i}$ avec $i \in Nd' - NNd'$, puis par le nombre maximum des $x_{k+2,i}$ et on continuera ainsi jusqu'à l'épuisement de Z unité de temps.

ETAPE 4 :

Calculer r le nombre de jours successifs du 2ème traitement dont leurs objets sont entièrement pris et ceci à partir de kième jour, on constate que r au mois est égal à 1.

Algorithme 5.1 : (Algorithme Marche en avant).

DEBUT

$k = 1, \quad r = d;$

TANT QUE $k \leq p-d-1$ FAIRE

DEBUT

Etape 1 ;

SI $r \geq d'$ ALORS $k := p-d$

SINON DEBUT

Etape 2;

Etape 3;

Etape 4;

$k = k+r;$

FIN ;

FIN ;

SI $r < d'$ ALORS DEBUT

SI $\sum_{i \in N_{p-d}} |p_i| \cdot b_i > rxs$ ALORS Etape 2;

FIN ;

FIN ;

III/ APPLICATION NUMERIQUE :

On considère que le nombre de jours de décalage $d = 2$;
Le nombre de jours de la période $P = 5$,
Il y a une seule chaine de 1er traitement.

Les parts sont les peintures $P_1, P_2, \dots, P_{12}$, donc $n = 12$ peintures.

Le temps de traitement d'un objet de la part P_i est $b_i = 1$ unité.
 $1 \leq i \leq n$

Le temps de transition si on passe d'un objet d'une part P_i à un objet de la part P_k est une constante c ne dépendant pas de la peinture.

Rappelons que l'on cherche à déterminer l'ordonnancement de 1er traitement.

Il y a trois chaines de 2ème traitement.

Le tableau suivant représente l'ordonnancement en type de 2ème traitement

jour \ chaine	1ère chaine	2ème chaine	3ème chaine
1)	$T_1 : 350$ $T_2 : 111$ $= 461$	$F_1 : 200$ $F_2 : 150$ $F_3 : 26$ $= 376$	$H_1 : 225$ $H_2 : 63$ $= 288$
2)	$T_2 : 89$ $T_3 : 372$ $= 461$	$F_3 : 254$ $F_4 : 116$ $= 370$	$H_2 : 78$ $H_3 : 200$ $H_4 : 7$ $= 285$
3)	$T_3 : 73$ $T_4 : 215$ $T_5 : 169$ $= 457$	$F_4 : 209$ $F_5 : 151$ $= 350$	$H_4 : 295$
4)	$T_5 : 231$ $T_6 : 220$ $= 451$	$F_5 : 367$	$H_4 : 48$ $H_5 : 130$ $H_6 : 200$ $H_7 : 25$ $= 403$
5)	$T_6 : 5$ $T_7 : 125$ $T_8 : 100$ $T_9 : 90$ $= 320$	$F_5 : 7$ $F_6 : 25$ $F_7 : 85$ $F_8 : 225$ $= 342$	$H_7 : 100$ $H_8 : 35$ $H_9 : 140$ $H_{10} : 145$ $= 420$

MATRICE DES BESOINS COMPATIBLE AVEC LES CHAINES 1, 2, 3.

part TYPE	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	P ₁₁	P ₁₂	TOTAL
T ₁			230							15		105	350
T ₂			77							78		45	200
T ₃			185							184		76	445
T ₄			80							87		48	215
T ₅			152							168		80	400
T ₆			82							97		46	225
T ₇			46							55		24	125
T ₈			34							45		21	100
T ₉						22		34		34			90
F ₁						56		67		77			200
F ₂						41		55		54			150
F ₃						41		82		95	62		280
F ₄						50		100		110	65		325
F ₅		180	180								105	60	525
F ₆		25		25									25
F ₇												60	85
F ₈								88		102	35		225
H ₁			50	85					25	10	55		225
H ₂			31	15				45		34		16	141
H ₃			125		55						20		200
H ₄			50	68			60	83		89			350
H ₅				82				39		9			130
H ₆	35		127	100			38						200
H ₇			25										125
H ₈	35			71				39					35
H ₉	30												140
H ₁₀	100				25		20						145
TOTAL	200	205	1347	573	80	210	118	632	25	1343	342	581	5656

2ème TRAITEMENT

-

1ère CHAÎNE

MATRICE X_1 -

part type	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	P ₁₁	P ₁₂	JOUR
T ₁ T ₂			230							15 66		105 45	1er jour
T ₂ T ₃			77 112							12 184		76	2ème jour
T ₃ T ₄			73 80							87		48	3ème jour
T ₅										89		80	

2ème TRAITEMENT

1

3ème CHAÎNE

MATRICE X_3

2ème TRAITEMENT - MATRICE $X = X_1 + X_2 + X_3$

part jour	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	P ₁₁	P ₁₂
1er jour			280	100		123		154	25	222	55	166
2ème jour			345	7	55	65		96		325	147	76
3ème jour			203	61			12	182		375	105	174
p-d + 1 ième jour	200	205	519	405	25	22	106	200	0	421	35	165

$Q_1 = 1125$

ORDONNANCEMENT 1er TRAITEMENT

ETAPE 1

OBJETS A TRAITER LES d PREMIERS JOUR DE 1er TRAITEMENT

S = 5656 : 5 = 1131

P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	P ₁₁	P ₁₂	TOTAL
		280	100		123		154	25	222	55	166	1125=Q ₁
			7		65		96		2	107	76	
			61				182			105	174	
					22					35	165	
		280	168		210		432	25	224	342	581	2262

TOTAL

1er jour (P₆, P₁₁, P₁₂) |P₆| + |P₁₁| + |P₁₂| = 1133 ; nombre de changement 2

2ème jour (P₉, P₈, P₄, P₉, P₁₀)

|P₉| + 432 + 168 + 280 + 224 = 1129 ; nombre de changement 4

MATRICE X réduite de 2 Lignes -

Objets non encore pris

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
1												
2												
3			195				12					
$p-d+1$	200	205	519	405			106	200				

$r=1$

$q_1=207$

Le reste de $p_3 + p_7 = 832 < s$ fin algorithme 5.1

Ordonnancement pour le $(d+2)$ ième jours $p_3 \rightarrow 714$

$|p_7| = 118$; 2 changements

$p_4 \rightarrow 299$

Ordonnancement pour le $(d+3)$ ième jour (le dernier)

$p_4 \rightarrow 106$

$p_8 \rightarrow 200$

$p_{10} \rightarrow 421$

$|p_1| = 200$

$|p_2| = 205$

4 changements

MATRICE X réduite d'une ligne

objets non encore pris

	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	P ₁₁	P ₁₂
1												
2			345		55					323		
3			203			12				375		
p-d+1	200	205	519	405	25	106	200			421		

r=1

Q₁=723

Etape 2. - Les objets à traiter de (d+1)ième jour de 1er traitement sont :

$$P_{10} \rightarrow 323 + 375 = 698$$

$$|P_5| = 80$$

$$P_3 \rightarrow 345 + 8 = 353$$

$$= 1131$$

$$\text{; nombre de changements } 2$$

TABLEAU RECAPITULATIF ORDONNANCEMENT DE 1er TRAITEMENT (Marche en avant)

	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	p ₉	p ₁₀	p ₁₁	p ₁₂	TOTAL
1er jour						210					342	581	1133
2ème jour			280	168				432	25	224			1129
3ème jour			353		80					698			1131
4ème jour			714	299			118						1131
5ème jour	200	205		106				200		421			1132
TOTAL	200	205	1347	573	80	210	118	632	25	1343	342	581	5656

TABLEAU RECAPITULATIF ORDONNANCEMENT DE 1er TRAITEMENT (marche en arrière)

	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	p ₉	p ₁₀	p ₁₁	p ₁₂	TOTAL
1						210					342	551	1133
2			625	100				223	25	156			1129
3					80			374		677			1131
4			505	473			118	35					1131
5	200	205	217							510			1132
TOTAL	200	205	1347	573	80	210	118	632	25	1343	342	581	5656

IV/ PROBLEME $n|m|1,m_1|(D1,T3),D2|C.1 -$

Ce problème peut être résumé ainsi, étant donné n parts, m types, un décalage fixé, quel est l'ordonnement de coût minimum pour les deux traitements.

On rappelle qu'il y a une chaîne de 1er traitement et m_1 chaînes de 2ème traitement travaillant en parallèle.

$$(m_1 \geq 1)$$

de plus le temps de transition entre :

deux objets de différente part est $c_{i,k} = c$ pour tout $i,k \in \{1, \dots, n\}$
 $i \neq k$

deux objets de différent type est $e_{j,k} = |d_j - d_k|$ pour tout $j,k \in \{1, \dots, m\}$
 $j \neq k$

Chaque chaîne de 2ème traitement admet deux ordonnancements minimum en temps de transition,
 ordonnancement selon l'ordre décroissant des d_j ,
 ordonnancement selon l'ordre croissant des d_j .

Par conséquent, le 2ème traitement admet 2^{m_1} ordonnancement minimum en temps de transition.

L'étude faite dans les chapitres II et III nous conduit à fixer l'ordonnement de 2ème traitement ; à chaque ordonnancement parmi les 2^{m_1} ordonnancement ; on peut trouver l'ordonnement (de 1er traitement) de transition minimum, admettant le nombre de jours de décalage le plus petit (algorithme 2.3.2), une fois cet ordonnancement trouvé, on propose d'augmenter le temps de transition de 2ème traitement en changeant l'ordre des types, pour diminuer le nombre de changements en parts dans le 1er traitement, (algorithme 5.1), on choisira l'ordonnement admettant le plus petit coût en temps pour les deux traitements.

A N N E X E

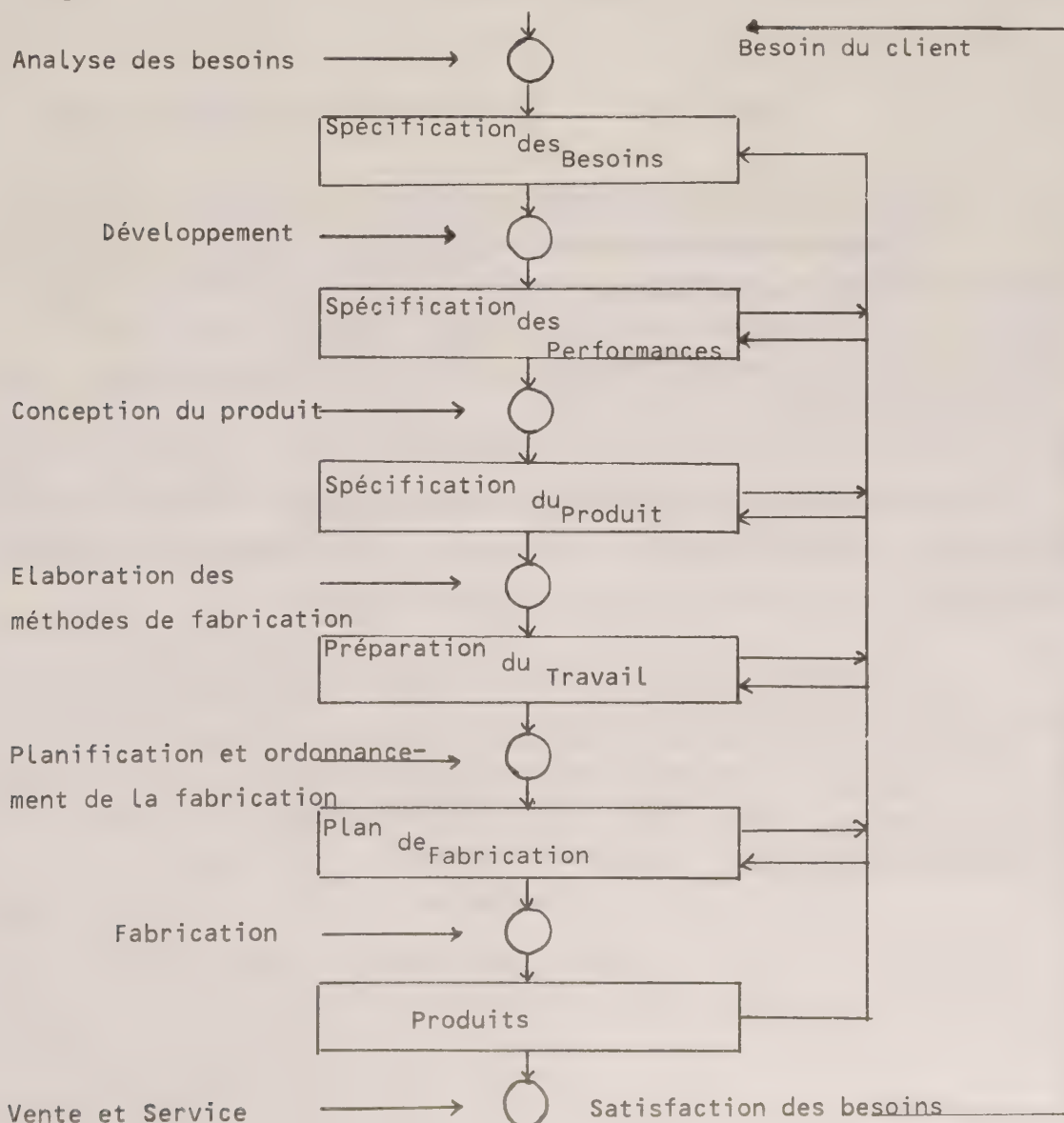
GESTION DE L'ORDONNANCEMENT

I - INTRODUCTION -

Le système de production comprend un certain nombre de fonctions que nous pouvons classer en trois grandes catégories [22] :

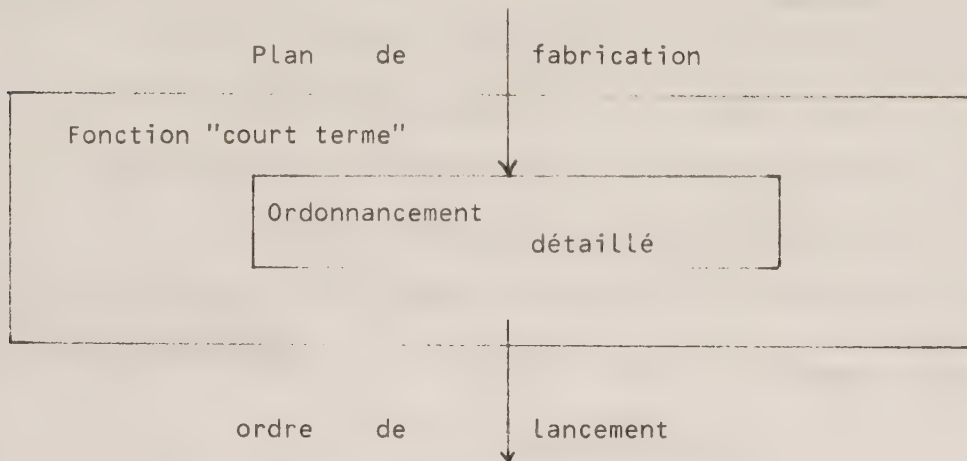
- celles relatives à la conception du produit,
- celles relatives à la préparation de la fabrication,
- celles relatives à la fabrication du produit.

La figure (A.I) montre les différentes phases d'un processus de production



Le système de conduite de production est habituellement décomposé en trois sous-systèmes, assurant les fonctions globales [22] :

- précision ou plan de production à long terme,
- planning à moyen terme,
- le court terme : (figure A.II) :



A partir du plan de fabrication, établi par la fonction calcul des besoins, on effectue un ordonnancement des opérations, celui-ci permet de connaître les dates de lancement en fonction des opérations dans l'atelier. A ce niveau, il est procédé à un ordonnancement à priori, opération par opération et chaîne par chaîne. Cette fonction trouve son prolongement dans le lancement et le suivi de fabrication.

Cependant des incidents surviennent fréquemment, incidents sur une chaîne, défaut d'approvisionnement ; il importe alors d'effectuer un suivi de fabrication pour réagir en temps réel à ces incidents.

La ligne directrice de cette thèse était l'ordonnancement à priori. L'annexe, qui s'en écarte, est une réflexion sur le suivi des traitements. Dans les paragraphes suivant nous reprenons quelques idées faisant partie de cours de Monsieur TREHEL (79-80 : S.M.I. Maîtrise, cours de système et T.D. de buretique [21]).

II - LIEUX CONCERNES -

Les lieux de l'entreprise concernée sont :

- la direction de fabrication, où on connaît la matrice des besoins et où se fait, pour le moment, l'ordonnancement,
- la chaîne de 1er traitement,
- la chaîne de 2ème traitement,
- le magasin de stock. C'est ce magasin qui va savoir quels objets manquent,
- le centre de traitement de l'information.

III - ORGANISATION CENTRALISEE -

On suppose que la direction reçoit les commandes et détermine la matrice de besoins pour une période donnée. Il la donne au centre de traitement de l'information qui établit les ordonnancements par ordinateur.

L'ordonnancement du 1er traitement est envoyé à la chaîne correspondante. L'ordonnancement du 2ème traitement est envoyé à la chaîne de 2ème traitement. Les deux ordonnancements sont envoyés au magasin de stock, ce dernier fait alors les commandes pour que tout soit prêt pour la date du début de la période.

A/ Incidents -

Les incidents que nous considérons sont les suivants :

- 1/ rupture de stock d'un objet donné (avant la période),
- 2/ non arrivage des objets prévus (en cours de période),
- 3/ impossibilité d'utiliser les objets arrivés (objets défectueux) (avant la période et en cours de période),
- 4/ incidents des chaînes (en cours de période),

5/ la direction de fabrication doit soit ajouter ou supprimer, soit ajouter et supprimer un nombre d'objets. (par exemple ajouter un groupe supplémentaire d'objets non prévus initialement pour raison commerciale ou supprimer un groupe d'objets prévus en fonction des moyens disponibles pour effectuer une tâche donnée pendant un temps t . (incident en cours de période)).

B/ Caractéristiques d'un ensemble d'objets donnés -

Les objets à traiter pour une période donnée sont caractérisés par :

1. Leur date de traitement :

- début de la période pour un traitement,
- fin de la période : p ième jour de 1er traitement,
les jours $p-d+1, p-d+2, \dots, p$ ième jour de 2ème traitement.

2. Leur temps de traitement : la durée d'exécution de ces objets dans le 1er et 2ème traitement peut être important (à définir par la direction) au moins pour une chaîne.

3. Leur date de livraison : une date de livraison avant la période, et une date de livraison en cours de la période si ces objets sont à traiter à la fin de la période d'un traitement.

4. Leur partition en parts et en types : en effet chaque objet (ou ensemble d'objets) est déterminé par une ligne dans la matrice des besoins :

	p_1	p_2		p_{n-1}	p_n
T	x_1	x_2		x_{n-1}	x_n

avec $x_i \geq 0$; T est le type, p_i est la i ème part.
 $1 \leq i \leq n$

donc tout changement dans la matrice de besoins consiste à appeler l'une des procédures suivantes :

- procédure ADD.($T, x_1', x_2', \dots, x_n'$), $x_i' \geq 0$; cette procédure consiste à ajouter des nouveaux objets.

pour $i=1, \dots, n$ faire $x_i := x_i + x_i'$.

- procédure SOUS($T, x_1', x_2', \dots, x_n'$), $x_i' \geq 0$; cette procédure consiste à soustraire le nombre d'objets à supprimer.

pour $i=1, \dots, n$ faire $x_i := x_i - x_i'$.

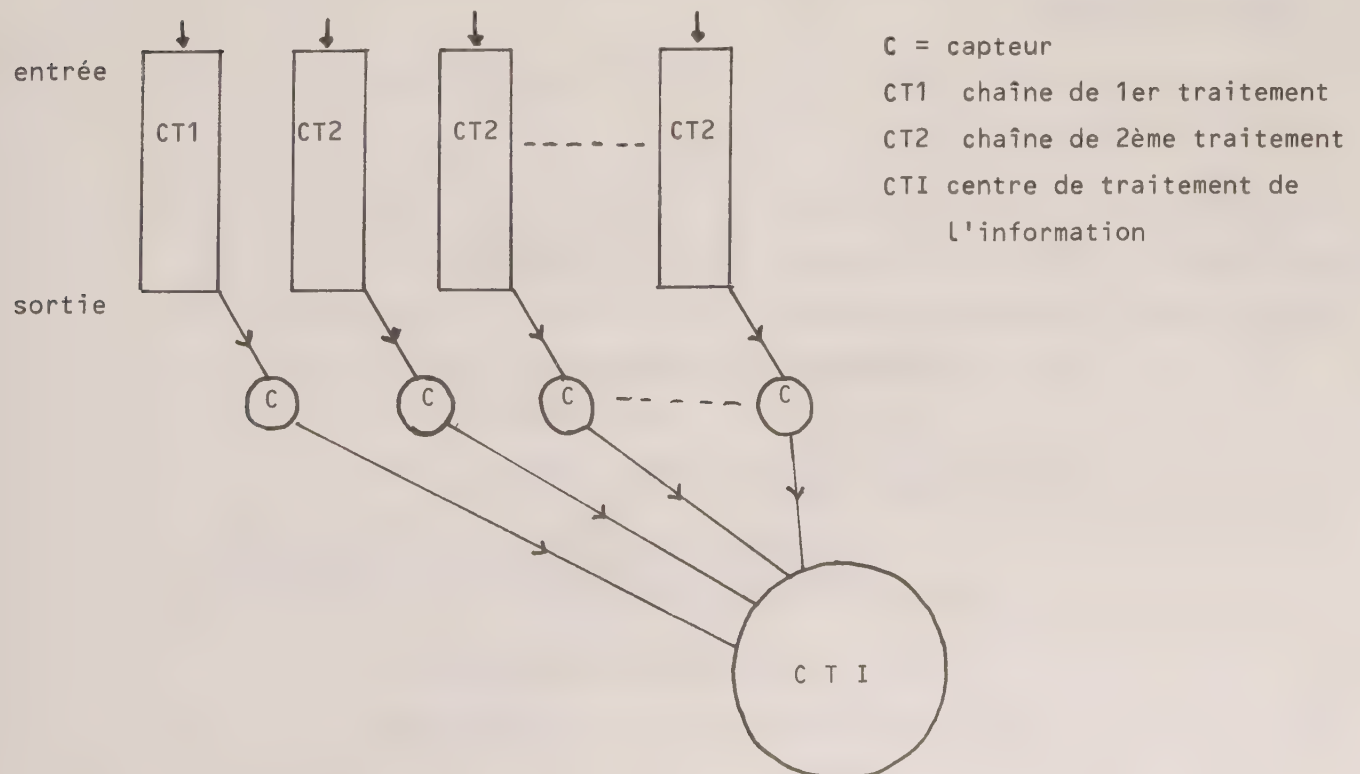
C/ dispositif à mettre en place -

Pour assurer une gestion efficace on propose de mettre en place le dispositif suivant :

- à la sortie de chaque chaîne de 1er et 2ème traitement un capteur relié au centre de traitement de l'information pour permettre à l'ordinateur de connaître à tout moment :

- . ce qui est passé en 1er traitement,
- . ce qui est passé en 2ème traitement,

donc on peut connaître l'état du stock entre les deux traitements. Signalons que chaque objet est accompagné d'une fiche d'identification.



Le programme de rattrapage que nous allons proposer est différent du programme à priori puisque une partie des objets a déjà été fabriquée.

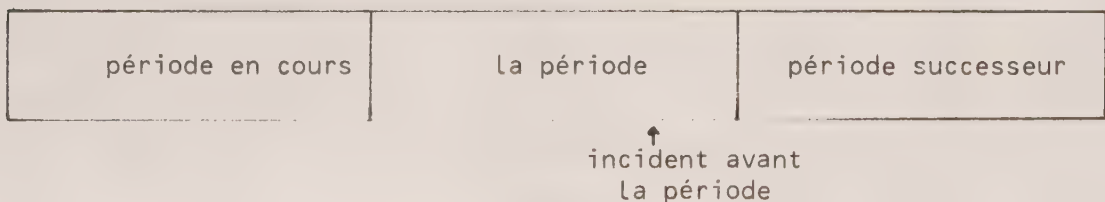
D/ Programmes de rattrapage -

Nous allons proposer deux programmes.

D.1. - Programme de rattrapage pour incident avant la période -

Cas de rupture de stock d'un objet donné, si ces objets admettent un temps de traitement non important pour toutes les chaînes, on les supprime de la matrice des objets à fabriquer durant la période, ou bien injecter dans cette période des objets de même type et de même part que la période successeur pour lesquels il y a plus de chance d'être approvisionné.

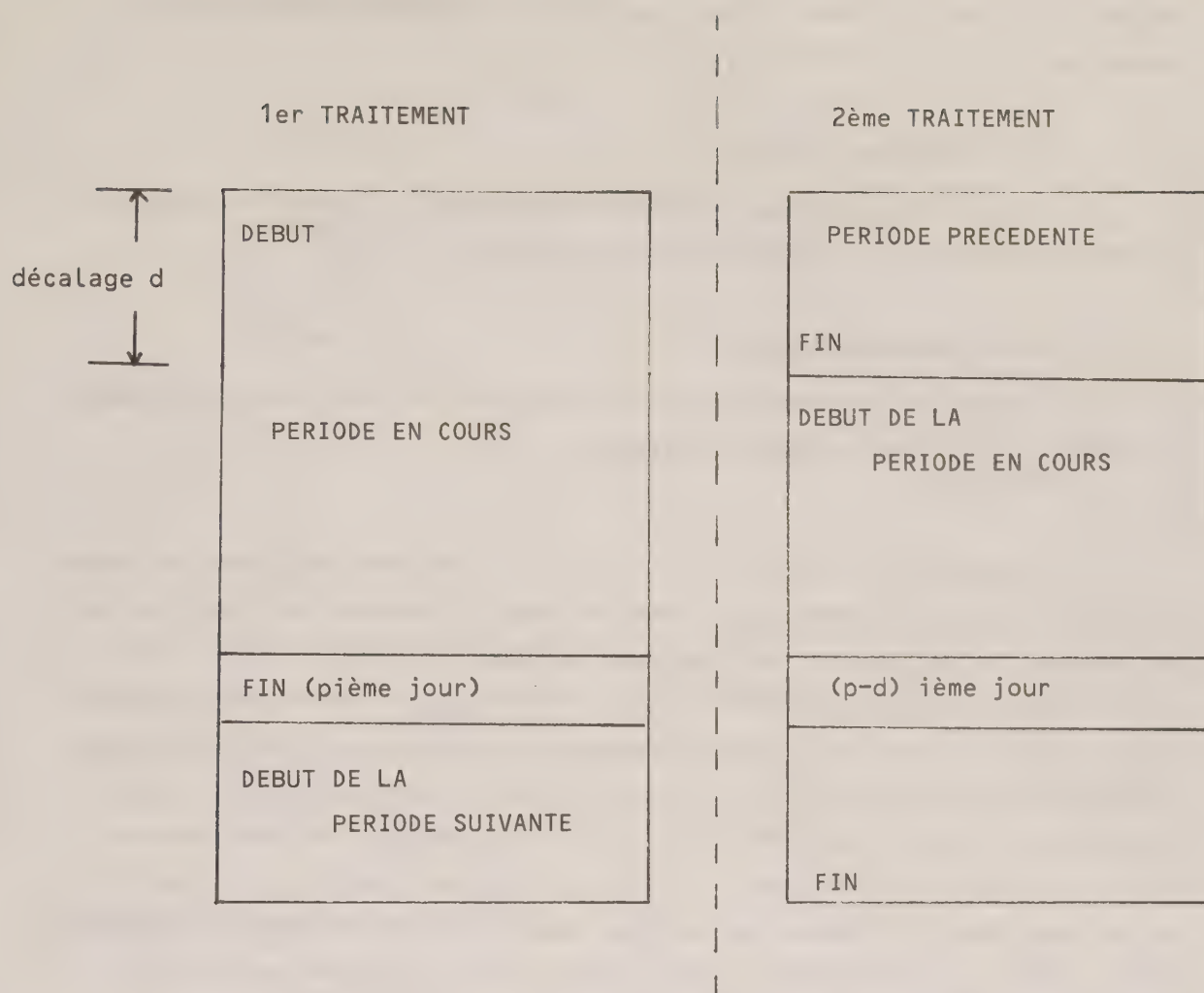
Par contre, si ces objets admettent un temps de traitement important au moins pour une chaîne, la solution est d'envoyer l'information à la direction de fabrication qui change la matrice des objets à fabriquer durant la période et on recommence à déterminer l'ordonnancement pour chaque traitement.



D.2. - Programme de rattrapage pour incident au cours de la période -

Ces incidents sont caractérisés par leur date :

- k ième jour de 2ème traitement pour la période en cours,
- h ième jour de 1er traitement pour la période suivante.



Pour assurer un suivi de traitement efficace au moment même de l'incident, nous pensons qu'il est nécessaire de connaître un certain nombre d'informations d'où l'intérêt des procédures suivantes sur chaque période.

1/ Procédure ST(p,k) -

Cette procédure détermine le stock d'objets entre les deux traitements (à partir des capteurs) à la date k ième jour de 2ème traitement (objets déjà passés dans le 1er traitement).

2/ Procédure ST((p+1),h) -

Elle détermine le stock d'objets pour la période suivante de la

période en cours à la date h ième jour de 1er traitement (objets déjà passés dans le 1er traitement).

3/ Procédure NH(p,k) -

Détermine le nombre d'hommes nécessaires, à la date k ième jour de 2ème traitement, et ceci pour chaque chaîne.

4/ Procédure NC(p,k) -

Détermine le nombre de chaînes nécessaires pour chaque traitement à la date k ième jour de 2ème traitement.

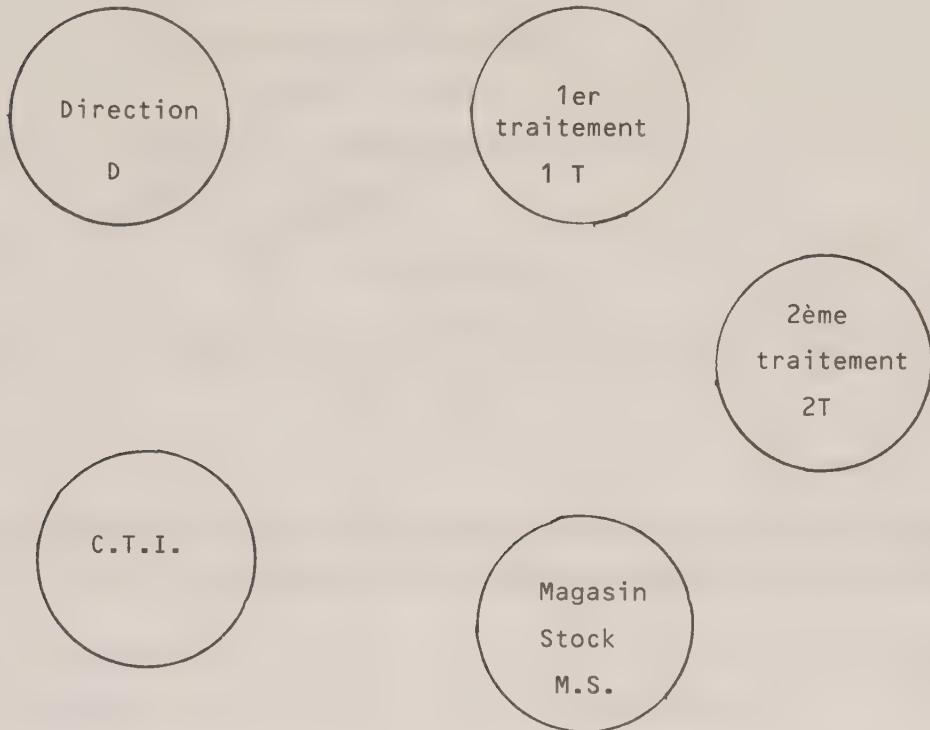
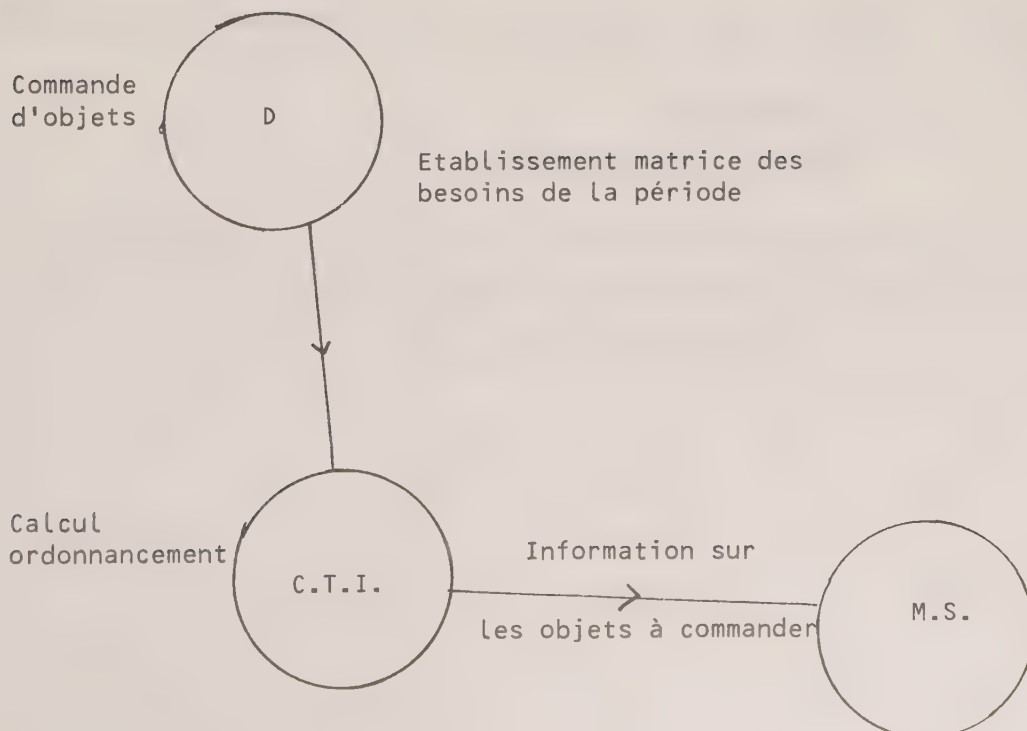
Ceci montre bien qu'il est inutile d'informer la direction chaque fois qu'il y a un incident, par contre on peut l'informer sur les résultats de journée, de période et seulement des incidents graves ; d'autre part les premières procédures montrent bien l'intérêt des transmissions automatiques. Ainsi il est inutile qu'on donne aux chaînes tout l'ordonnancement de la période alors que celui-ci risque d'être souvent modifié en cours de période. D'après les caractéristiques d'un ensemble d'objets (date de traitement, date de livraison), il est possible de commencer le travail de la période même s'il manque des objets, dans la mesure où ceux-ci ne doivent servir qu'en fin de période. Il peut alors y avoir incident en cours de période pour non arrivage des objets prévus ou incidents de chaînes.

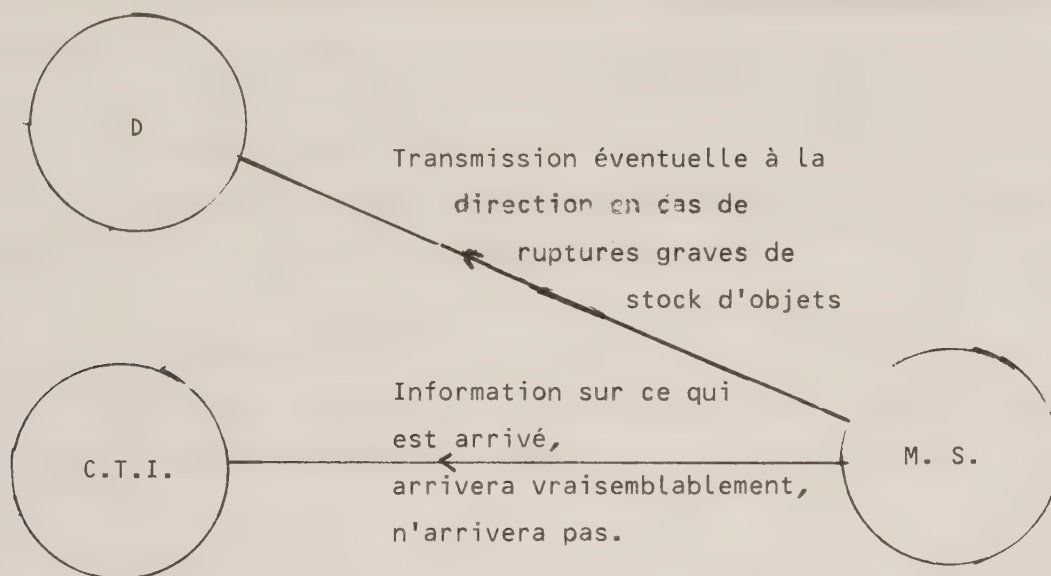
Dans ce cas, la chaîne de 1er traitement (ceci pour tout k ième jour de 1er traitement) dispose de deux ordonnancements : l'ordonnancement de temps de transition minimum et l'ordonnancement ne respectant pas le principe de travail par part (à cause du décalage fixé), algorithme 5.1. En effet, grâce aux procédures $ST(p,k)$ ou $ST(p+1,h)$, on peut injecter des objets (de la période suivante ou bien de jours suivants) dans la période en cours, tel que le temps de transition sur l'ensemble de deux traitements soit le plus petit possible.

D'autre part, tout changement d'ordonnancement pour chaque chaîne doit tenir compte des quantités d'objets, chaînes et hommes nécessaires.

IV/ ORGANISATION REPARTIE -

D'après le paragraphe II, les lieux concernés sont :

Travail début -

Réponse du Magasin de stock -Information des chaines de traitement concernant les résultats d'une journée ou de la période précédente et réponse de C. T. I.

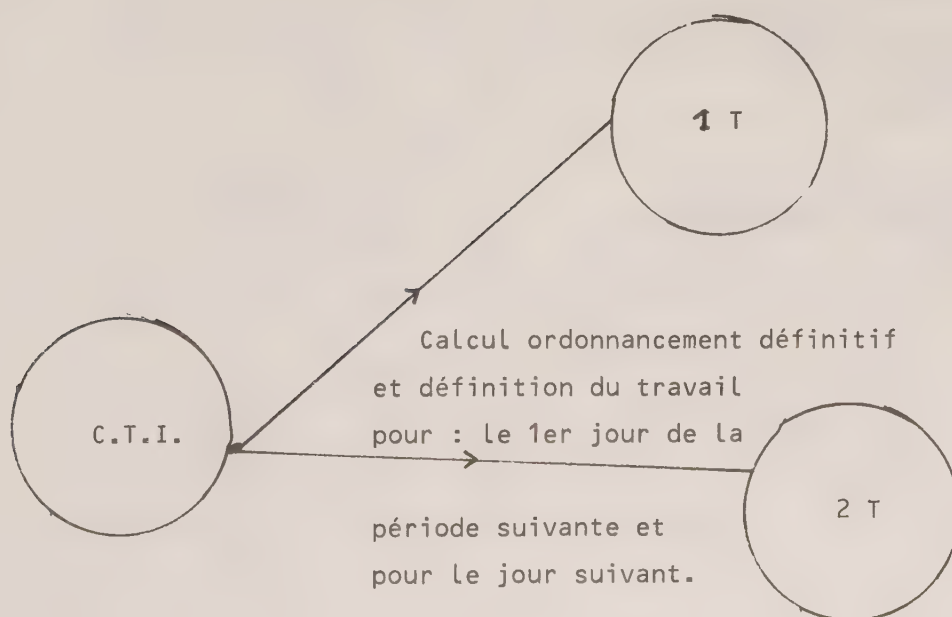
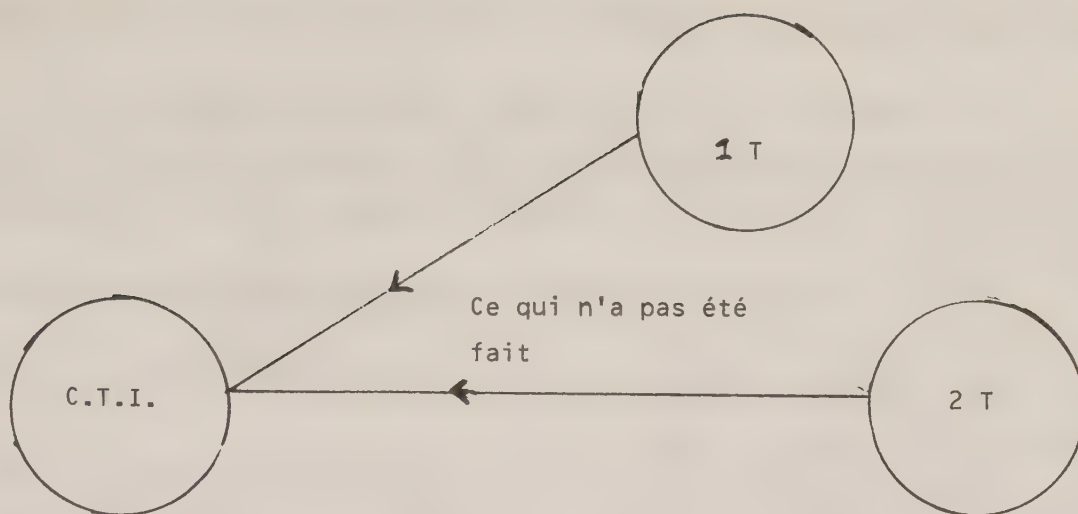
Chaque fin de journée ou de période, les chaines de traitement indiquent :

- ce qu'elles ont fait (confirmation des données du capteur automatique).
- ce qui n'a pas été fait.

De même, le magasin de stock a indiqué ce qui lui manque pour le travail prévu :

- le lendemain,
- sur la période suivante.

Le centre de traitement d'information fait alors la réorganisation et donne aux chaines la définition du travail du lendemain.



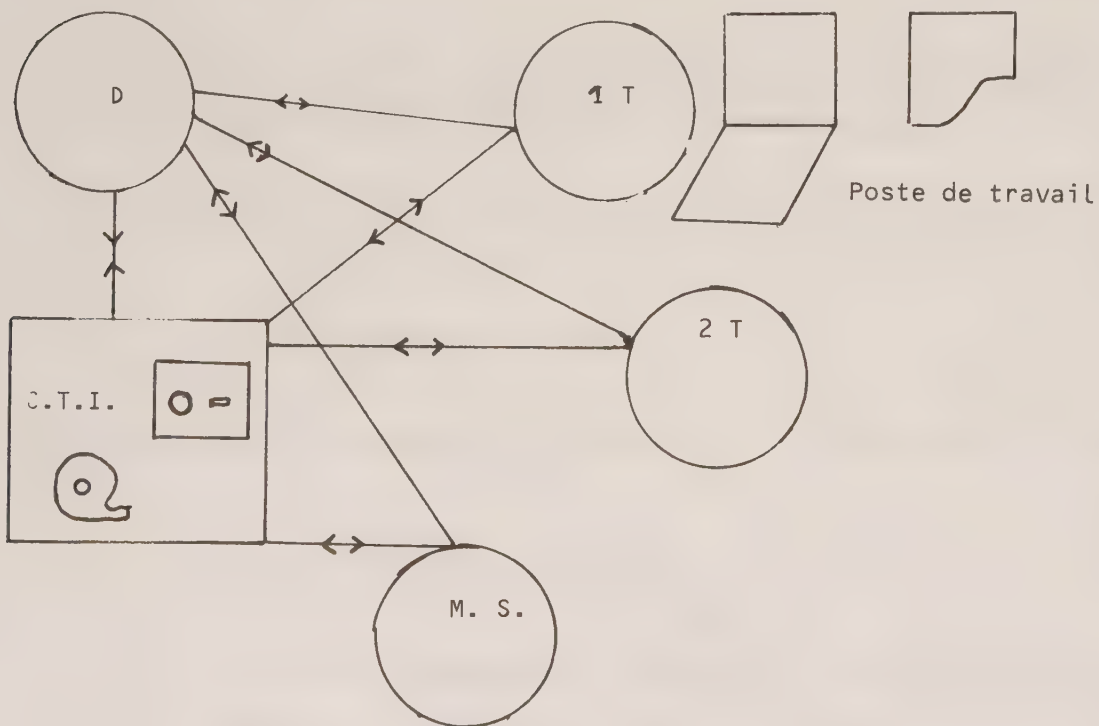
Proposition de structure du système -

On propose que chaque site dispose d'un clavier émetteur, d'un écran récepteur et d'une imprimante, pour des problèmes de synchronisation ; on peut admettre que la Direction dispose d'un écran récepteur correspondant à chaque émetteur.

Il est indispensable de garder la copie du travail réalisé chaque jour en 1 T et 2 T.

Pour les autres transactions, il est possible, mais pas toujours nécessaire d'en garder une copie.

Les données concernant la période en cours et la période suivante sont stockées sur disque, par contre l'information sur les autres périodes peut-être stockée sur bande magnétique. On propose le réseau simple suivant :



Voyons maintenant les transactions entre sites -1) Ecran de 1 T et 2 T -

En fin de journée, 1 T et 2 T indiquent ce qu'ils ont fait, l'écran enregistre l'information transmise.

Ordonnancement demandé	Partie réalisée
T 8 : 256	T 8 : 256
T 9 : 325	T 9 : 320

On remarque que si un objet stocké entre les deux traitements n'est pas pris à la date k ème jour de 2ème traitement c'est qu'il est encore disponible à la date $(k+1)$ ième jour de 2ème traitement.

2) Ecran de Magasin de stock -

M. S. indique à C. T. I., les dates d'obtension prévues.
Si ces dates sont passées, c'est qu'il les a déjà en stock.

Affichage de ce qui est demandé.	Affichage des dates d'obtension prévues.
A 8 : 1207	25.03.80
A 10 : 1325	15.03.80
A 11 : 25	18.03.80

CONCLUSION -

A la lecture de ce document on se rend compte que nous avons abordé le problème d'ordonnancement de production sur deux traitements successifs avec temps de transition dans le cas :

- d'une chaîne de 1er traitement ($n_1 = 1$),
- plusieurs chaînes de 2ème traitement ($m_1 \geq 1$),

pour une matrice au temps de transition constant ou symétrique.

Par contre, le cas avec une matrice au temps de transition quelconque reste à envisager, ainsi que le cas où le nombre de chaînes de 1er traitement est strictement supérieur à 1.

BIBLIOGRAPHIE -

- [1] M. TREHEL "Programme de lancement le plus économique en peinture et en montage fini".
Note 2, le 14.11.1977, L.A.N.I., BESANCON.
- [2] M. TREHEL "Programme de lancement le plus économique en peinture et en montage fini".
Note 3, le 16.01.1978, L.A.N.I., BESANCON.
- [3] S.M. JOHNSON "Optimal tow and three stage production schechules with setup times included".
Naval. Res. logist. quart 1. (61-68), 1954.
- [4] J.K. LENSTRA, A.H.G. RINNOOY KAN, B.J. LAGEWEG
"A general Bounding schema for the permutation flow-shop problem".
Opns. Res. Vol. 26, n° 1, (53-67), 1978.
- [5] R.W. CONWAY, W.L. MAXWEL, L.W. MILLER
"Theorey of scheduling".
Addison Wesley, 1967.
- [6] B. JULLIEN "Différentes approches logicielles pour la résolution des problèmes combinatoires en temps réel".
Thèse de 3ème cycle, TOULOUSE, 1976.
- [7] A.H.G. RINNOOY KAN
"Machine scheduling problems - Classification, Complexity and Computation".
Thèse de Doctorat, Nijhoff the HAGUE, 1976.
- [8] J.K. LENSTRA "Sequencingue by Enumerative Methods".
Thèse de Doctorat, Mathematical Centrum, AMSTERDAM, 1976.

- [9] J. LABETOULLE "Modélisation des systèmes informatiques et application aux réseaux d'ordinateurs".
Thèse de Doctorat, PARIS SUD, 1978.

- [10] P. NEPOMIASTCHY
" Résolution d'un problème d'ordonnancement à ressource variable".
R.A.I.R.O., Vol. 12, n° 3, (249-261), 1978.

- [11] L. NEMETI "Sur le problème d'ordonnancement dans la fabrication en série".
R.I.R.O., n° 11, (47-60), 1968.

- [12] J. CARLIER "Ordonnancement à contraintes disjonctives".
R.A.I.R.O., Vol. 12, Novembre 1978.

- [13] HIROSHI KISE, JOSH ICHIDE I BARAKI, HISASHI MINE
"A solvable case of the one machine scheduling problem with ready and Due Time".
Opns. Res. Vol. 26, n° 1, (121-126), 1978.

- [14] HAMILTON EMMONS
"One Machine sequencing to minimize certain Functions of job Tordiness".
Opns Res., Vol. 17, (701-715), 1969.

- [15] J.M. MOORE "AN n Job ONE Machine sequencing Algorithm for Minimizing the number of laste jobs".
Mang. Science, Vol. 15, n° 1, (102-109), 1968.

- [16] JANTINDER N.D. GUPTA
"A Heuristique algorithme for the flow-shop sheduling problem".
R.A.I.R.O., Vol. 10, n° 6, Juin 1976.

- [17] M.L. SMITH, S.S. PANWALKAR and R.A. DUDEK
• "Flow-shop sequencing probleme with ordered processing Time matrices".
Manag. Science, Vol. 17, n° 1, (544-549), 1975.
- [18] FENNELL BURNS et J. ROOKER
"A special case of the 3 x n flow-shop problem".
Naval. Res. Log. Quart. 22, (811-817), 1971.
- [19] Wladzimierz SZWARC
"Optimal elimination method in the m x n flow-shop scheduling problem".
Opns. Res., 21, (1250-1259), 1973.
- [20] Wladzimierz SZWARC
"Special cases of the flow-shop problem".
Naval. Res. Log. Quart. 24, (483-492), 1977.
- [21] M. TREHEL
"Cours de système et T.D. de burotique".
S.M.I., Maîtrise, 1979-1980.
- [22] G. LACOSTE
"Automatisation de la conduite de production".
C.N.A.M., Centre associé de BESANCON, Laboratoire d'automatique, 1979-1980.

